NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A241 815

DTIC
ELECTE
OCT 22 1991
S B D

THESIS

ESKAPE/CF: A KNOWLEDGE ACQUISITION TOOL FOR EXPERT SYSTEMS USING COGNITIVE FEEDBACK

by

James W. Connor, Jr.

March 1991

Thesis Advisor: Kishore Sengupta

Approved for public release; distribution is unlimited.

91-13653

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Administrative Science Dept. Naval Postgraduate School; 6b. OFFICE SYMBOL (if applicable): Code 37	7a. NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code): Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION; 8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
ESKAPE/CF: A KNOWLEDGE ACQUISITION TOOL FOR EXPERT SYSTEMS USING COGNITIVE FEEDBACK (U)

12. PERSONAL AUTHOR(S)
Connor, James, W. Jr.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Master's Thesis	FROM 09/89 TO 03/91	March 1991	89

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Expert Systems, Knowledge Acquisition, Cognitive Feedback

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The major bottleneck in the construction of expert systems is the time-consuming process of acquiring knowledge from experts. Automated knowledge acquisition tools have demonstrated the ability to reduce the time required to construct expert system knowledge bases and are supported by both knowledge engineers and experts. However, due to limitations in their underlying psychological paradigms, existing tools may not be well-suited to extracting semantic or procedural knowledge from an expert.

This thesis designs and implements an Expert System Knowledge Acquisition and Policy Evaluation tool using Cognitive Feedback (ESKAPE/CF), based on Lens model techniques which have demonstrated effectiveness in capturing policy knowledge. The system is designed to be used interactively by an expert to reduce the historically lengthy interactions with a knowledge engineer. Additionally, the use of cognitive feedback techniques should enable the system to capture expertise that has heretofore been unobtainable by existing knowledge acquisition tools.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Kishon Sengupta	22b. TELEPHONE (Include Area Code): (408) 646-3212 / 22c. OFFICE SYMBOL: AS/SE

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

Approved for public release; distribution is unlimited.

ESKAPE/CF: A Knowledge
Acquisition Tool for Expert Systems
Using Cognitive Feedback

by

James W. Connor, Jr.
Lieutenant Commander, United States Naval Reserve
B.S., University of Illinois, 1979

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1991

Author: _____
James W. Connor, Jr.

Approved by: _____
Kishore Sengupta, Thesis Advisor

_____
Tung Bui, Second Reader

_____
David R. Whipple, Jr., Chairman
Department of Administrative Sciences

ABSTRACT

The major bottleneck in the construction of expert systems is the time-consuming process of acquiring knowledge from experts. Automated knowledge acquisition tools have demonstrated the ability to reduce the time required to construct expert system knowledge bases and are supported by both knowledge engineers and experts. However, due to limitations in their underlying psychological paradigms, existing tools may not be well-suited to extracting semantic or procedural knowledge from an expert.

This thesis designs and implements an Expert System Knowledge Acquisition and Policy Evaluation tool using Cognitive Feedback (ESKAPE/CF), based on Lens model techniques which have demonstrated effectiveness in capturing policy knowledge. The system is designed to be used interactively by an expert to reduce the historically lengthy interactions with a knowledge engineer. Additionally, the use of cognitive feedback techniques should enable the system to capture expertise that has heretofore been unobtainable by existing knowledge acquisition tools.

DTIC COPY INSPECTED 4

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist Special
A-1

TABLE OF CONTENTS

I. INTRODUCTION 1

A. GENERAL 1

B. BACKGROUND 2

C. PROBLEM DESCRIPTION 3

D. THESIS OBJECTIVE 4

E. SCOPE 4

F. ORGANIZATION OF THE STUDY 4

II. LITERATURE REVIEW 6

A. INTRODUCTION 6

B. PROBLEMS IN ACQUIRING KNOWLEDGE FROM EXPERTS 7

1. The Nature of Expertise 7

2. Expert Learning 8

3. Expert Decision Making 8

C. KNOWLEDGE ACQUISITION 9

1. Types of Knowledge 10

2. Decision Tasks 12

3. Direct Knowledge Acquisition 12

4. Indirect Knowledge Acquisition 13

5. Automated Knowledge Acquisition 13

6. Limitations of Knowledge Acquisition . 14

D. COGNITIVE FEEDBACK 15
1. The Lens Model 15
2. Single Systems 16
3. Cognitive Feedback 17
4. Presentation of Cognitive Feedback . . . 17
E. SUMMARY 18

III. ESKAPE/CF: A TOOL FOR KNOWLEDGE ACQUISITION . . . 20
A. System Overview 20
1. Terminology 21
a. Cue 21
b. Judgment node 21
c. Root judgment 21
d. Child 22
e. Sibling 22
f. Cue weights (beta weights) 22
g. Subtree 22
2. ESKAPE/CF Hardware and Software Requirements 22
3. ESKAPE/CF Session Overview 23
4. ESKAPE/CF Application Areas 24
5. Program Limitations 25
a. Decision Cues 25
b. Discrete Decision Cues and Judgment Nodes 25

B. SAMPLE KNOWLEDGE ACQUISITION SESSION 26
1. Program Initiation and Operation 27
2. Initializing Root Judgment Value 30
3. Adding New Cues 30
4. Exploding Cues 34
5. Correlating Cues 40
6. Evaluating Cues 43
7. Feedback 46
a. Cue Relationships with Decision Weights 46
b. Decision Weights 49
c. Function Forms 49
8. Viewing the Knowledge Base 49
9. Generating Sample Cases 54
10. Editing Cues 57
11. Deleting Cues 57
12. Moving Cues 57
13. Saving the Knowledge Base 63
14. Loading Knowledge Bases 68
15. Clear Tree 68
C. SUMMARY 68

IV. CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH . 71
A. CONCLUSIONS 71

B. FUTURE RESEARCH 72
1. ESKAPE/CF System Enhancements 72
a. Linear Logistic Analysis 72
b. X-Windows Compatibility 72
c. Applying ESKAPE/CF Knowledge to a Working Expert System 73
d. Expand the ESKAPE/CF Data Keeping Facilities 73
2. Assessing ESKAPE/CF Validity 73
a. Applicable Task Areas 73
b. System Usability 73
c. Comparison of Results using Different Experts 73
d. Comparison of Results with Other Models 74
3. ESKAPE/CF Extensions 74
a. Learning Complex Tasks 74
b. Combining Expert Knowledge 74
c. Study the Decision Processes of Experts 75

APPENDIX. SAMPLE USER TRACE 76

LIST OF REFERENCES 77

INITIAL DISTRIBUTION LIST 81

I. INTRODUCTION

A. GENERAL

One of the most critical aspects in the building of expert systems is the formulation of knowledge bases used by those systems. Despite considerable advances in computer technology, the major bottleneck in the construction of knowledge bases continues to be the time-consuming process of acquiring knowledge from experts (Olson & Rueter 1987).

Automated knowledge acquisition tools have demonstrated the ability to reduce the time required to construct expert systems (Boose 1985). But, due to limitations in their underlying psychological paradigms, current tools may not be well-suited to extracting semantic or procedural information from an expert (Boose 1985; Patterson 1990).

This research proposes an **E**xpert **S**ystem **K**nowledge **A**cquisition and **P**olicy **E**valuation tool using **C**ognitive **F**eedback, **ESKAPE/CF**. ESKAPE/CF is based on Lens model techniques which have demonstrated effectiveness in capturing policy knowledge (Balzer, Doherty, & O'Connor 1989). The ESKAPE/CF prototype model evolved from initial design descriptions proposed by Charles Patterson (1990).

B. BACKGROUND

Expert systems can be defined as a computer-based system consisting of a user interface, an inference engine and a "knowledge base" (McNurlin & Sprague 1989, 448). Expert systems are tailored to solve specific problems and are becoming prevalent in such diverse areas as engine failure diagnosis, tax planning, space shuttle crew schedules, and law and regulation interpretation (Olson & Rueter 1987; McNurlin & Sprague 1989, 450; Patterson 1990).

The difficult and time-consuming process of eliciting knowledge from experts is often referred to as "knowledge engineering." Although expertise for knowledge bases can also be acquired from books, databases, reports, etc., it is most often obtained by direct interaction with an expert (Patterson 1990). Traditional methods of knowledge engineering such as interviews and protocol analysis may be of limited usefulness because the expert can not always articulate how his or her decisions are made (Cooke & McDonald 1988; Boose 1985). This tedious process of obtaining relevant information from experts has often driven knowledge engineers to becoming experts themselves (Olson & Rueter 1987).

Insufficient numbers of trained knowledge engineers coupled with the possibility of losing knowledge in the transfer process has led to the increased use of interactive systems directly by an expert (Shaw & Gaines 1988). These

automated tools have demonstrated the capability to shorten project development time and thus reduce the bottleneck in the construction of expert systems. Additionally, the success of such knowledge elicitation tools has challenged the cost effectiveness of the knowledge engineer as an intermediary in the knowledge acquisition process (Shaw & Gaines 1988). Unfortunately, since no single model can capture all levels or types of expertise (Olson & Rueter 1987), careful evaluation of available knowledge acquisition tools and techniques is required to match the tool with the particular application (Kitto & Boose 1989).

C. PROBLEM DESCRIPTION

The background section highlights one of the major problem areas with knowledge acquisition tools. Specifically, the tool selected must be tailored to the individual expert system application and the individual expert whose knowledge is to be captured.

Use of the proper knowledge acquisition tool will minimize the disparity between the knowledge elicited and the actual knowledge held by the expert. Furthermore, any translation errors that might be introduced during the traditional knowledge engineering process will be diminished (Patterson 1990).

D. THESIS OBJECTIVE

This thesis has as its primary objective, the construction of a prototype knowledge acquisition tool, **ESKAPE/CF**, based on an earlier study conducted at the Naval Postgraduate School that determined initial high level specifications (Patterson 1990).

E. SCOPE

The scope of this thesis is limited to the construction of the prototype knowledge acquisition tool, **ESKAPE/CF** and generation of a sample knowledge acquisition session for demonstration purposes. Since this thesis is a follow-on to the work of Charles Patterson (1990), the literature review basically consists of a summary of his research with minor additions.

F. ORGANIZATION OF THE STUDY

This thesis, excluding the introduction and conclusion, is divided into two major sections. The first main section, Chapter II, is basically a summary of the research conducted by Charles Patterson. The information presented is necessary for reader to understand the theory upon which the **ESKAPE/CF** model is based. This chapter highlights knowledge acquisition techniques, principles of cognitive feedback, expert judgement strategies, and a limited discussion of current knowledge acquisition tools.

The second main section, Chapter III, consists of a description of the **ESKAPE/CF** model including a "walk-through" of a sample knowledge acquisition session. Using this chapter as a guide or "Users Manual", an expert will be able to use the **ESKAPE/CF** system to generate a knowledge base.

II. LITERATURE REVIEW

A. INTRODUCTION

The task of constructing a knowledge base for an expert system is multi-faceted and interdisciplinary. To capture expert knowledge, one must first understand how an expert makes a decision and how that decision knowledge, or expertise, is represented. The proper method must then be chosen to elicit that knowledge from the expert. Only when the relevant knowledge is extracted can the actual construction of a knowledge base begin.

Decision theory, behavioral science, and cognitive science are among the disciplines that contribute to the understanding of expertise and expert decision making (Balzer, Doherty, & O'Connor 1989; Gaines 1987). Additionally, they lay the framework around which knowledge elicitation techniques are developed. A basic understanding of the aforementioned techniques and underlying theories is essential when constructing any knowledge acquisition system. This section is a broad overview of these topics. A detailed discussion is contained in Patterson (1990).

B. PROBLEMS IN ACQUIRING KNOWLEDGE FROM EXPERTS

Shanteau (1988) describes an expert as one who is "considered by colleagues to be the best at making decisions." Whether one uses the preceding description or identifies an expert on the basis of test scores or "self-proclamation," difficulties can arise when trying to acquire that expert's knowledge. Additionally, the nature of that knowledge or expertise is viewed differently by different disciplines.

1. The Nature of Expertise

Cognitive psychologists view expertise as domain specific and maintain that experts develop problem solving techniques that are rooted in that domain of expertise. Therefore, an expert's performance is often lessened outside his or her field of specialization. (Shanteau 1990) Additionally, as they gain experience, experts tend to rely on "automated" decision processes, similar to pattern recognition, in their thinking (Shanteau 1990; Larkin, McDermott, Simon, & Simon 1980). However, no matter how much experience an expert has accumulated, expert thinking is not infallible.

Shanteau (1990) concludes that "experts are inadequate decision makers." Research indicates (Einhorn 1974; Shanteau & Gaeth 1981; Carroll & Payne 1976) that experts are often unreliable and inaccurate (Shanteau 1990). Experience tends to make an expert more confident, but is not necessarily

related to enhanced decision making abilities, improved accuracy, or improved consistency (Meehl 1954; Shanteau 1990). Furthermore, it appears that experts themselves are unaware of the aforementioned shortcomings (Shanteau 1990).

2. Expert Learning

The acquisition of expertise can be viewed as a three stage transformation of knowledge representation. Fitts and Posner (1967) identify the first of these stages as the "cognitive stage," where the expert memorizes facts required to perform a given task. At this stage, the expert can easily articulate how he or she makes a decision or performs a task. (Shanteau 1990; Anderson 1982)

The second stage is the "associative stage," where the expert begins forming decision rules between elements he or she has memorized in the previous stage. (Shanteau 1990; Anderson 1982)

The final stage is the "autonomous stage," where the skill acquisition is complete. At this stage, the acquired skills become almost automatic and the expert may have difficulty in relating exactly how he or she performs the given task. (Shanteau 1990; Anderson 1982)

3. Expert Decision Making

When making a decision, an expert must select inputs from a myriad of cues and decide which of those cues to assimilate and which to ignore (Phelps & Shanteau 1978;

Einhorn 1974). The expert then clusters similar cues to reduce the complexity of the information processing task, assigns weights to each of the cues, and combines them to achieve a final decision (Einhorn 1974). Although one would expect an expert to use all relevant data, empirical data suggests that they often do not. It has been shown that some experts use as few as three cues while others can integrate as many as nine to eleven cues (Phelps & Shanteau 1978). Additionally, some experts use these cue combinations as the basis for heuristic decision making (Shanteau 1990).

Unfortunately, experts are often poor at accurately assessing the weights and combinations of cues used in their decision making processes (Einhorn 1974). In fact, experts are often unable to articulate exactly what cues they consider when making a decision (Cooke & McDonald 1988; Boose 1985; Gaines 1987; Shanteau 1988). Furthermore, their expertise may not be understandable or expressible in language. It may also be inapplicable, incomplete, irrelevant, or incorrect (Gaines 1987). Knowledge acquisition techniques must compensate for these shortcomings.

C. KNOWLEDGE ACQUISITION

Research has shown that "expertise is domain specific" (Shanteau 1990). Although some decision *strategies* have been used in different decision situations, expertise tends to be tailored to cognitive tasks depending on the particular

problem (Shanteau 1990; Shanteau 1988). These tasks, in turn, may require manipulation of different types of knowledge that the expert possesses (Patterson 1990). This section highlights the types of decision tasks confronting an expert, the types of knowledge used in those decisions, and an overview of current techniques in eliciting that knowledge from an expert.

1. Types of Knowledge

Although several methods of classifying knowledge exist in literature, none is universally accepted. (Patterson 1990). One widely used methodology was proposed by McGraw and Riner (1987) which classified knowledge into four basic types: procedural, declarative, semantic, and episodic (Patterson 1990).

Declarative knowledge is "surface level information that experts can verbalize" (Patterson 1990). This differs from procedural knowledge in that the expert is aware of its existence and can articulate it. Declarative knowledge is generally easy to acquire (McGraw & Riner 1987).

Procedural knowledge includes deeply ingrained skills which may include automatic responses to stimuli. Experts develop this knowledge over time, have great difficulty in articulating or even identifying it (McGraw & Riner 1987; Patterson 1990).

Semantic and episodic knowledge are the two theoretical components of long term memory. Semantic knowledge is difficult for experts to express because it reflects cognitive organization, structure, and representations and includes such memories as vocabulary, concepts, definitions, and interrelationships between facts. A knowledge engineer must effectively capture semantic knowledge to ensure a viable expert system. (McGraw & Riner 1987; Patterson 1990)

Episodic knowledge represents experiential and autobiographical information that the expert chunks into episodes. This information is difficult to extract because the knowledge is stored in chunks and the expert may not be aware of individual knowledge entities or how it affects his or her decision-making processes. (McGraw & Riner 1987; Patterson 1990)

An expert's knowledge, therefore, undergoes a transformation -- from declarative knowledge, that the expert is aware of, to procedural knowledge which the expert may not even be able to identify. This evolution of knowledge is akin to the transformation of skills from the "cognitive stage" to the "autonomous stage", as discussed in the previous section. It is for this reason that experts have so much difficulty in expressing knowledge -- procedural knowledge is made

autonomous, leaving the expert often unaware of the exact steps in the decision making process. (Shanteau 1990; McGraw & Riner 1987; Anderson 1982)

2. Decision Tasks

As with knowledge types, there is no universally accepted theory that can categorize all types of problem-solving tasks. However, these tasks can be associated with one of two broad categories summarized by Kitto and Boose (1989): *analysis* (interpretive) tasks and *synthesis* (constructive) tasks (Patterson 1990). Examples of analysis tasks include such areas as diagnosis and debugging while synthesis tasks include evaluation, planning, and scheduling. These two task areas can be combined to yield *analysis-synthesis* tasks which include control, monitoring, and repair (Patterson 1990).

3. Direct Knowledge Acquisition

Direct knowledge acquisition techniques, such as interviews or questionnaires, rely on experts to articulate their knowledge. This may be of limited usefulness in extracting procedural, semantic or episodic knowledge because experts may not be able to "express what they know" in language (Gaines 1987; Patterson 1990). Furthermore, transcription and analysis of an expert's responses are extremely time-consuming, increasing both the development time and cost of an expert system.

4. Indirect Knowledge Acquisition

Indirect knowledge acquisition techniques rely on various psychological paradigms to avoid relying on an expert's articulation and introspection (Patterson 1990). These methods rely on the knowledge engineer to determine the appropriate psychological model "*a priori*" to ensure that the selected method matches the task and type of knowledge required.

5. Automated Knowledge Acquisition

One of the most common paradigms used in automated knowledge acquisition is Kelly's Personal Construct Theory (1955) using repertory grid techniques (Boose 1985). These programs require the expert to assess decision cues and their interrelationships to determine decision rules (Boose 1985). Repertory grids are used to classify and cross-reference real-world objects. Because people can usually discern between differences and similarities (Garg-Janardan & Salvendy 1988), the expert is presented with groups of three objects and asked to indicate how two of the items are alike and yet different from the third. This characterization of objects can be on a bi-polar scale (either it has a trait or it doesn't) or on an ordinal scale (normally 1-5) (Shaw & Gaines 1988). Prototype expert systems constructed using Personal Construct techniques have been developed in as little as two hours (Kitto & Boose 1989).

Repertory grid techniques work best at identifying traits and relationships for analysis class or structured problems, but are not as effective in obtaining causal knowledge -- when or how information is used in a decision-making task (Boose & Bradshaw 1988; Boose 1985). Furthermore, Personal Construct psychology procedures do not guarantee that sufficient knowledge will be acquired to solve a specified problem (Boose & Bradshaw 1988).

6. Limitations of Knowledge Acquisition

Since no single model can capture all levels or types of information (Olson & Rueter 1987), careful evaluation of available knowledge acquisition tools and techniques is required to match the tool with the particular application (Kitto & Boose 1989). Although several knowledge acquisition techniques have been useful in extracting declarative knowledge, there are no satisfactory methods for consistently extracting semantic or procedural knowledge (Patterson 1990).

The survey of knowledge acquisition techniques by Patterson (1990) indicated the need for an automated knowledge acquisition tool that could elicit semantic and procedural knowledge. Techniques based on cognitive feedback (CFB) and the lens model have demonstrated potential in such tasks (Balzer, Doherty, & O'Connor 1989).

D. COGNITIVE FEEDBACK

This section provides an overview of the lens model, cognitive feedback techniques and possible applications of cognitive feedback.

1. The Lens Model

Brunswick's lens model (1955) addresses decision-making under uncertainty (Patterson 1990) and the integration of information from multiple decision cues (Einhorn, Kleinmuntz, & Kleinmuntz 1979). "The model can be viewed as an individual judging an event or object (criterion), which cannot be directly perceived, through a lens of cues" (Patterson 1990).

Libby (1981) further refined the model into three separate elements: the criterion event, the task environment (X_1, X_2,...,X_k), and the judge's estimate. The lens model summarizes the relationships between these elements (see Figure 1) (Patterson 1990).

The cognitive system of an individual judge is denoted by the right side of the lens model. The degree that a judge relies upon individual cues is measured by the relationship between the cue (X_i) and the judgement (Y_s). This *utilization coefficient* or *beta weight* may be positive or negative. If a cue is ignored by the judge in the decision making process, the beta weight will be zero. The left side, or task system,

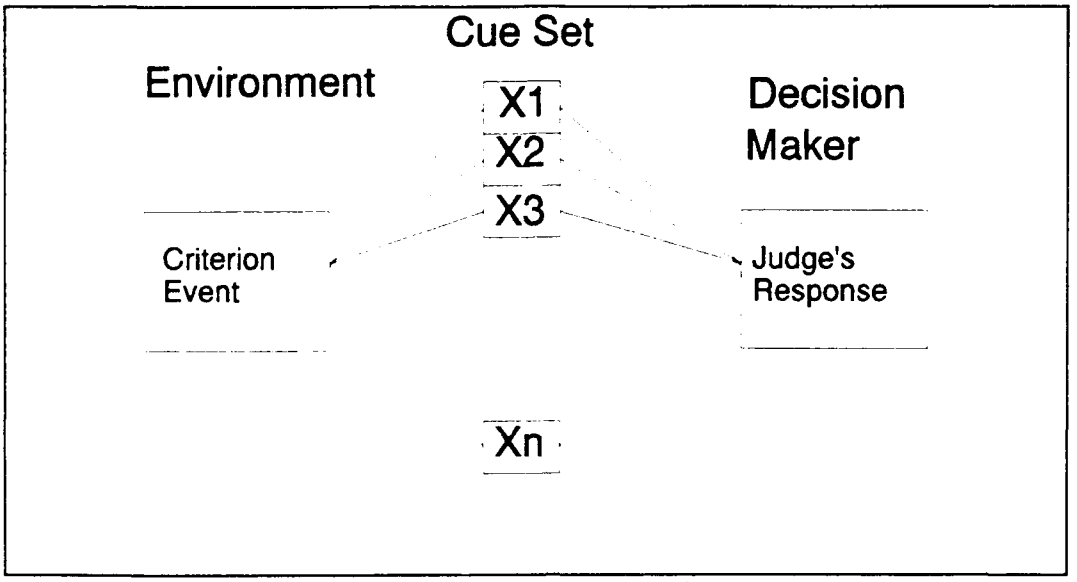

Figure 1. Simplified Lens Model. (Libby 1981)

of the lens model is similar to the right side, as required by the principle of parallel concepts. (Patterson 1990)

2. Single Systems

The *single system paradigm*, as shown in Figure 2, involves analysis of only the cognitive side of the lens model, whereas a double system paradigm (Figure 1) involves interaction between both the cognitive and task sides. This single system view allows analysis of a set of cues and judgements by linear regression methods (Brehemer 1979). Research has demonstrated that the decision processes of an individual can be simulated by this model, using the beta weights for individual cues as the coefficients for the regression variables (Patterson 1990).

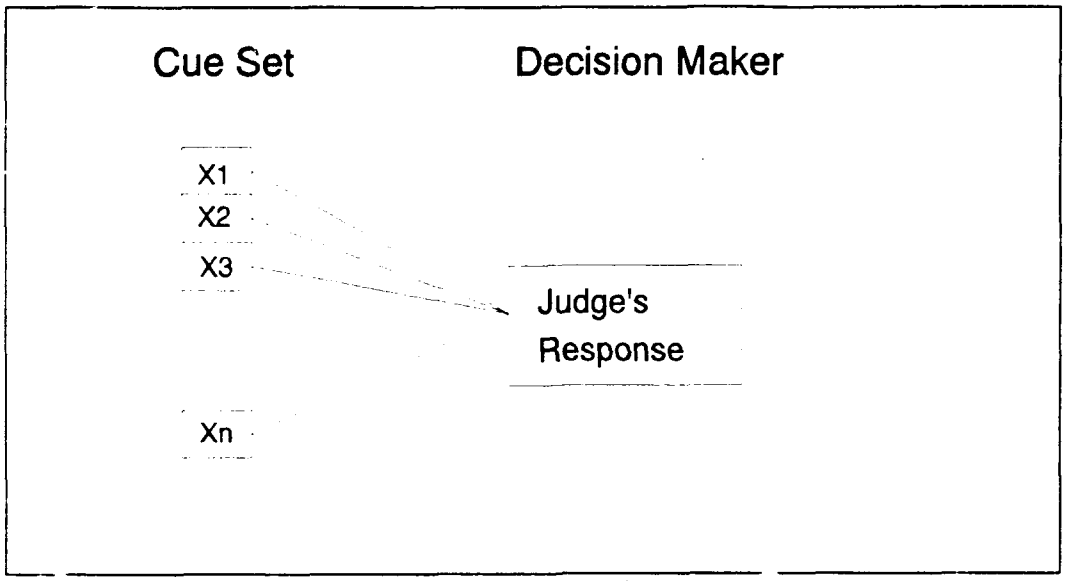

Figure 2. Single System Case. (Libby 1981)

3. Cognitive Feedback

Cognitive information returned to a decision maker can be used to gain insight into that decision maker's value system in a given environment. Research indicates that individuals often cannot verbalize their decision policies. When they do verbalize, those descriptions are often inaccurate descriptions of their decision policies. Cognitive feedback can therefore be used to provide experts with an understanding of their decision policies if they are unknown. (Balzer, Doherty, & O'Connor 1989; Patterson 1990)

4. Presentation of Cognitive Feedback

Cognitive feedback can be presented in various formats. Because individuals differ in how they best absorb cognitive information, graphically or verbally, any cognitive

aid should incorporate both methods to maximize effectiveness. *Function forms* have been used to graphically display the relationships between individual cues and the ultimate judgement. Another method is to present individual *beta values* as obtained from the single system lens model regression equation. These beta values are usually displayed graphically, but textual and verbal formats are also used. (Patterson 1990)

E. SUMMARY

Single-system cognitive feedback techniques can be used for assigning weights to the various cues on which experts base their decisions. These policy capturing techniques have been successfully used as decision aids for judgment analysis (Balzer, Doherty, & O'Connor 1989) and can often characterize an expert's judgments better than experts themselves (von Winterfeldt 1988). Insights, via cognitive feedback, into the policies that experts use to make decisions could also offset the fact that experts cannot always articulate their knowledge (Boose 1985) or express specific weights for individual policy cues (Doherty & Balzer 1988). Use of these techniques, therefore, may prove better suited for use in eliciting knowledge in synthesis class problems than current paradigms such as Personal Construct psychology.

Since delay in presenting cognitive feedback to an individual reduces the effectiveness of that feedback (Patterson 1990), a computer-based cognitive aid which can instantaneously provide textual or graphical feedback could be an ideal judgment evaluation tool. The following chapter describes the **E**xpert **S**ystem **K**nowledge **A**cquisition and **P**olicy **E**valuation using **C**ognitive **F**eedback (**ESKAPE/CF**) model. This prototype knowledge acquisition tool incorporates the timely and flexible presentation of computer graphics, coupled with the proven effectiveness of cognitive feedback techniques.

III. ESKAPE/CF: A TOOL FOR KNOWLEDGE ACQUISITION

A. System Overview

The Expert System Knowledge Acquisition and Policy Evaluation using Cognitive Feedback system is a prototype knowledge acquisition tool which uses cognitive feedback techniques and lens model algorithms to elicit and record expert knowledge. It can also be used by experts to clarify their decision making processes. As defined by its specifications (Patterson 1990), ESKAPE/CF provides for direct interaction with the user, a sound theoretical paradigm, and may be broadly applicable across various domains.

Using the ESKAPE/CF system, the expert enters the various cues upon which his or her decision is based. These cues, in turn, may have further sub-cues. As cues are entered, the expert can specify correlations between pairs of cues to indicate any interrelationships. This type of scaling requires the expert to make an internal judgment which is more effective in capturing decision policies than external introspection (Cooke & McDonald 1988).

Once all cues and correlations have been entered, the user is presented with sample cases and asked to assess the cues and make appropriate judgments. After the judgments are made,

the expert is provided with cognitive feedback in the form of lens model beta weights and function forms based on judgments made in the test cases. The user can then accept the current cues/weights or edit the cues and run more test cases.

The ESKAPE/CF interface is designed so that an expert can expand and test the knowledge base without restrictions. Help facilities are available and the system notifies the user in the case of incomplete or inconsistent information. The interface is designed to be flexible, to minimize differences with the expert's cognitive style which can reduce resistance to using the system (Hoffman 1989; Olson & Rueter 1987).

1. Terminology

The following terms are used throughout the program explanations:

a. Cue

Value that an expert considers when making a judgment. An expert may consider several cues in the decision making process.

b. Judgment node

A judgment (or cue) whose value is determined by evaluating decision cues.

c. Root judgment

The ultimate decision being captured by the ESKAPE/CF system.

d. Child

A cue that is used by a judgment node in determining a decision. A judgment node may have several child cues.

e. Sibling

Two cues used by the same judgment -- both children of the same judgment node. Siblings are the only cues which can be correlated.

f. Cue weights (beta weights)

The relative percentage an expert places on an individual cue when making a decision. These values are calculated by the ESKAPE/CF program during the cue evaluation process.

g. Subtree

Group of cues headed by a judgment node which consists of all children, grandchildren, great-grandchildren, etc..

2. ESKAPE/CF Hardware and Software Requirements

The ESKAPE/CF system was developed on Sun/Sun compatible workstations. To provide "reasonable" system response times, a workstation running ESKAPE/CF should be rated at 20 MIPS or greater. Machines, slower than 20 MIPS, that were used in initial system development proved very slow in processing more than four decision cues due to extensive matrix calculations (Press and others 1990, 39-46).

The ESKAPE/CF system is written in "C" and was developed using Unix/SunOS version 4.0.3. The program uses SunWindows to generate the user interface and Sun CGI to provide some of the graphical presentations. A complete listing of required libraries and header files are contained in file definitions.h and the program makefile.

3. ESKAPE/CF Session Overview

The ESKAPE/CF system is designed so that the user can choose which tasks he or she wishes to perform. The program generates appropriate help messages to ensure that the expert does not choose a function if any prerequisites have yet to be completed. A typical ESKAPE session might proceed as follows:

1. Enter user name.
2. Define ROOT Judgment.
3. Create decision cues with ADD CUE.
4. Create any subcues required using EXPLODE CUE and ADD CUE as required.
5. Correlate decision cues as needed with CORRELATE CUES.
6. Evaluate all judgment nodes with more than two decision cues using EVALUATE CUE.
7. View the accumulated knowledge base using KNOWLEDGE BASE.
8. Generate sample case to verify the captured judgment policies using GENERATE CASES.
9. Save the knowledge base using SAVE FILE or SAVE AS and quit the ESKAPE Program.

Follow-on sessions and other ESKAPE/CF options include:

1. Load previously saved knowledge bases using LOAD FILE.
2. Edit previously defined cues with the EDIT CUE.
3. Delete previously defined cues with the DELETE CUE.
4. Move decision cues within the knowledge base using MOVE CUE.
5. Reevaluate decision cues and generate new cases.

4. ESKAPE/CF Application Areas

Research has indicated that cognitive feedback may be applicable to various problem solving techniques and may be best suited to inference tasks. It may also be used as a training tool in complex learning situations such as anti-submarine warfare tactics or nuclear reactor operations. (Patterson 1990) Other possible application areas discussed in literature (Patterson 1990, Balzer, Doherty, & O'Connor 1989; Hammond, McCleland, & Mumpower 1980) include:

1. Software estimation
2. Personnel evaluation
3. Economic forecasting
4. Battlefield situation assessment
5. Medical diagnosis
6. Hardware diagnosis
7. Security risk analysis

5. Program Limitations

The limitations "built into" the ESKAPE/CF system were intended to somewhat simplify the prototype program. These limitations can be lessened by varying parameters in the program or adding more efficient computational routines.

a. Decision Cues

Although Phelps and Shanteau (1978) indicate that some experts can use up to eleven decision cues, the ESKAPE/CF program limits the maximum number of decision cues to seven. This limit was established to place a fixed upper bound on storage space and computational time. Additionally, seven cues appeared a sufficient nominal value because other research indicates the cognitive limit for most individuals is five to seven items (Miller 1956). The maximum number of decision cues can be raised (or lowered) by simply changing one parameter in the ESKAPE/CF program.

b. Discrete Decision Cues and Judgment Nodes

By using the linear regression techniques offered by the lens model, one must assume that the possible values of a judgment and its decision cues are continuous in nature. Unfortunately, discrete or categorical decision variables are not continuous and therefore regression analysis may only provide a rough approximation of the beta values for these cues.

If the judgment node is continuous and the decision cues are discrete, *analysis of covariance* (or regression analysis with dummy variables) can be used to provide accurate analysis. Similarly, if the judgment node is discrete, *linear logistic* or *logit* models should be used instead of multivariable linear regression. (Fienberg 1977, 2-4) Incorporation of these models into the ESKAPE/CF program is left for future upgrades.

B. SAMPLE KNOWLEDGE ACQUISITION SESSION

One of the application areas previously discussed is the realm of software estimation. Vicinanza (1990) proposes a model for software cost estimation based on several factors such as *size*, *program attributes*, *personnel attributes* and *project attributes*. These factors are then broken down into further subcues for use by the model. For example, program complexity (CPLX), performance (PERF), and reliability (RELY) are some of the subcues for the factor *program attributes*.

The cues illustrated in Vicinanza (1990) are used to demonstrate how a knowledge base could be built using the ESKAPE/CF program. It should be noted that both continuous and discrete decision variables are used in the Vicinanza (1990) model. Therefore, as previously discussed, the linear regression routines in the prototype ESKAPE/CF model provide only approximations to beta values.

1. Program Initiation and Operation

The session begins with the user entering his or her name, up to fourteen characters, as shown in Figure 3. This is used to create a unique user trace file which will contain a listing of all actions taken by the user along with an appropriate timestamp. A sample user trace file is included in the Appendix.

The program is designed to be used with a mouse for selection tasks and the keyboard for entering textual or numeric data. When entering text or numbers, a carriage return is required to alert the program that an input has been made. When selecting a button or item, the user "points" the mouse at the desired item and presses the left mouse button. Inputs or selections are verified by the program and appropriate error messages are generated as required. Context sensitive help is also available to the user at any time by selecting the help button with the mouse.

The availability of a particular function is denoted by the appearance of its associated function button. As shown in Figure 4, the ESKAPE/CF screen is broken up into four major sections (the title block is just that). The **System Message** panel displays acknowledgement or error messages generated by the program. The **File Information** panel displays the name of the current expert file and contains buttons for file manipulation. The **ESKAPE Control Panel** contains all of the

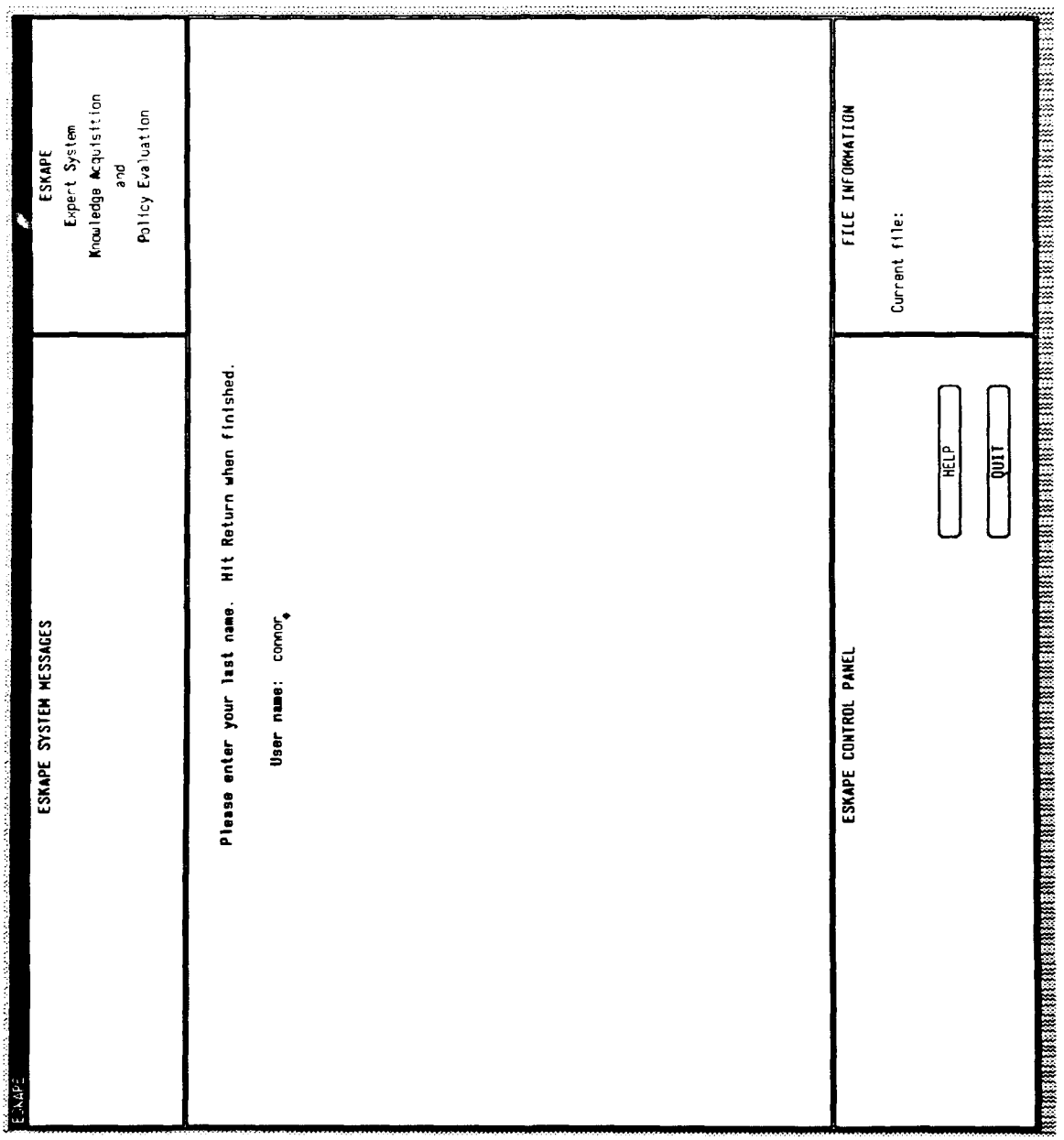

Figure 3. Entering user name.

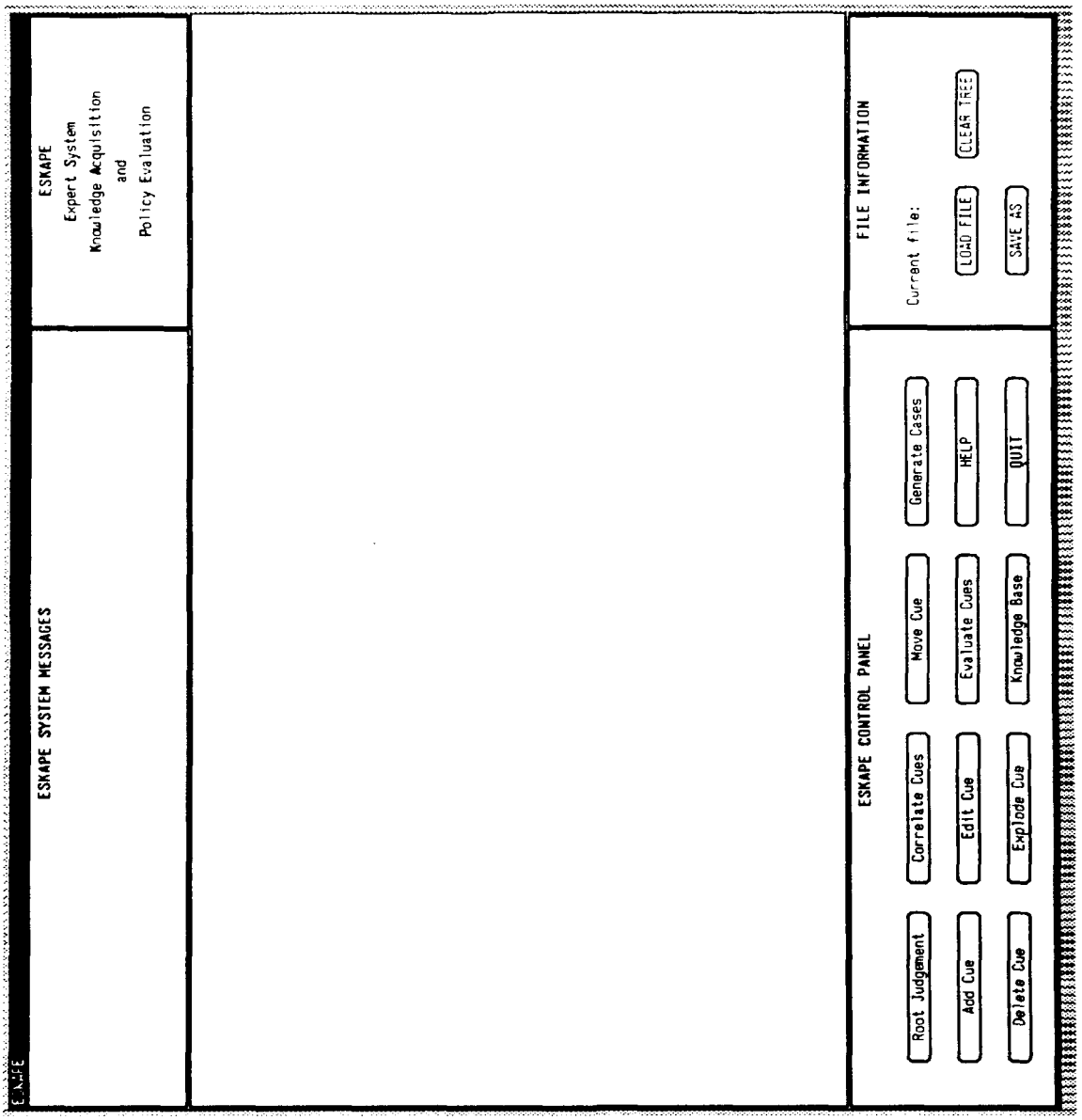

Figure 4. Generic ESKAPE/CF screen.

buttons used to perform cue creation, manipulation, and evaluation. The large central **Working Panel** is used to present displays and selections to the user.

2. Initializing Root Judgment Value

The first task an expert should accomplish is to set the type and possible values of the root judgment node. Figure 5 depicts the cue edit screen after selecting the ROOT Judgment button on the ESKAPE control panel. The cue type is selected by selecting the desired cue type, either **Integer**, **Float**, or **Categorical**. Integer and float cue ranges are then specified by entering the minimum and maximum values for that cue, as depicted by Figure 5. However, the user may enter up to ten discrete values for a categorical cue. Once the appropriate values for the cue are entered, the user should select the SAVE CUE DATA button.

3. Adding New Cues

After setting the root judgment, the user is ready to add the first decision cue by selecting the ADD CUE button on the ESKAPE control panel. The first cue in the knowledge tree is always a decision cue of the root judgment. Figure 6 shows the first decision cue *size* being entered as a float type.

After saving the *size* cue, the user is shown a selection screen similar to Figure 7. Here, the user selects a cue where he or she desires to add a sibling. In this

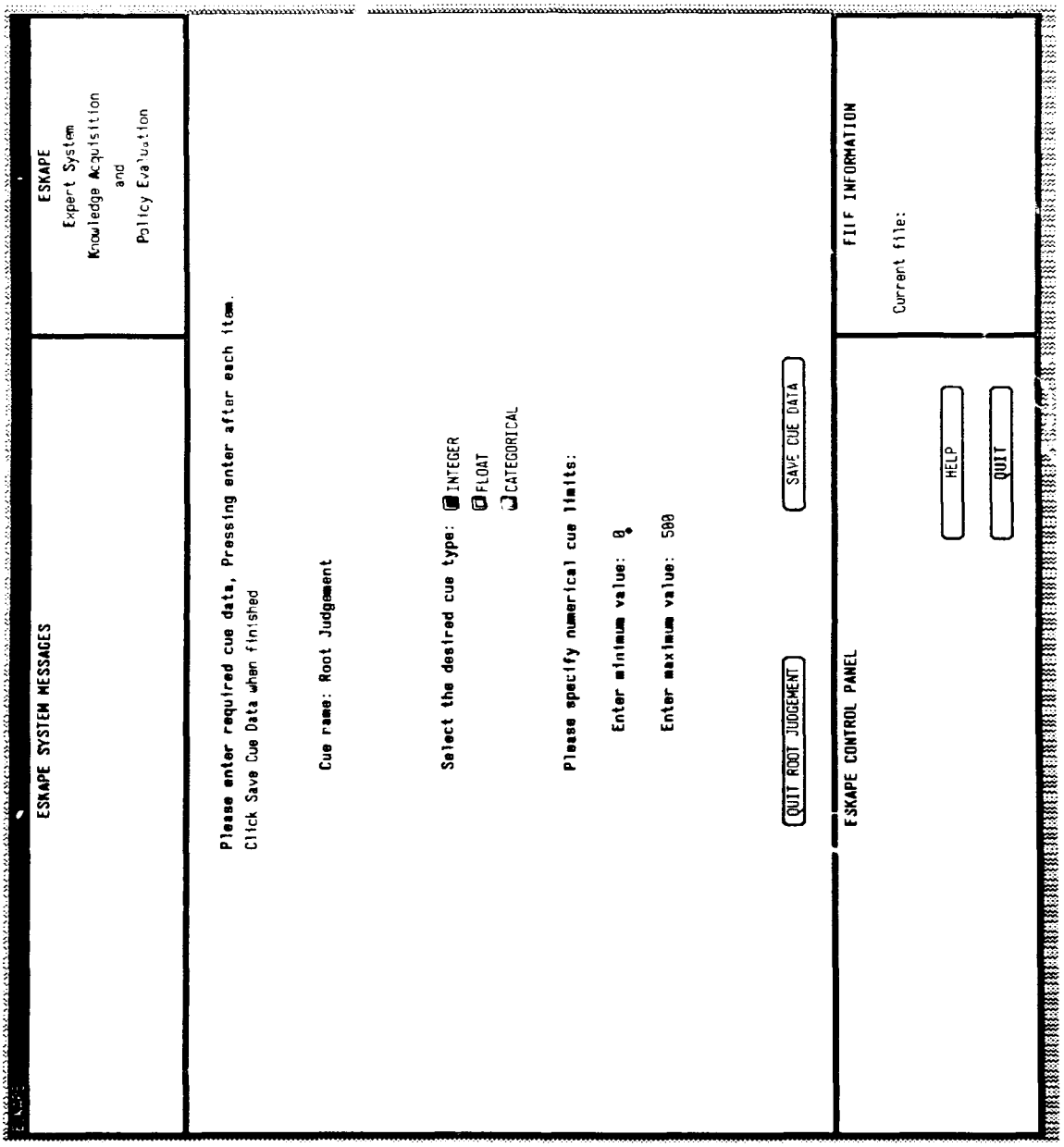

Figure 5. Creating the root judgment.

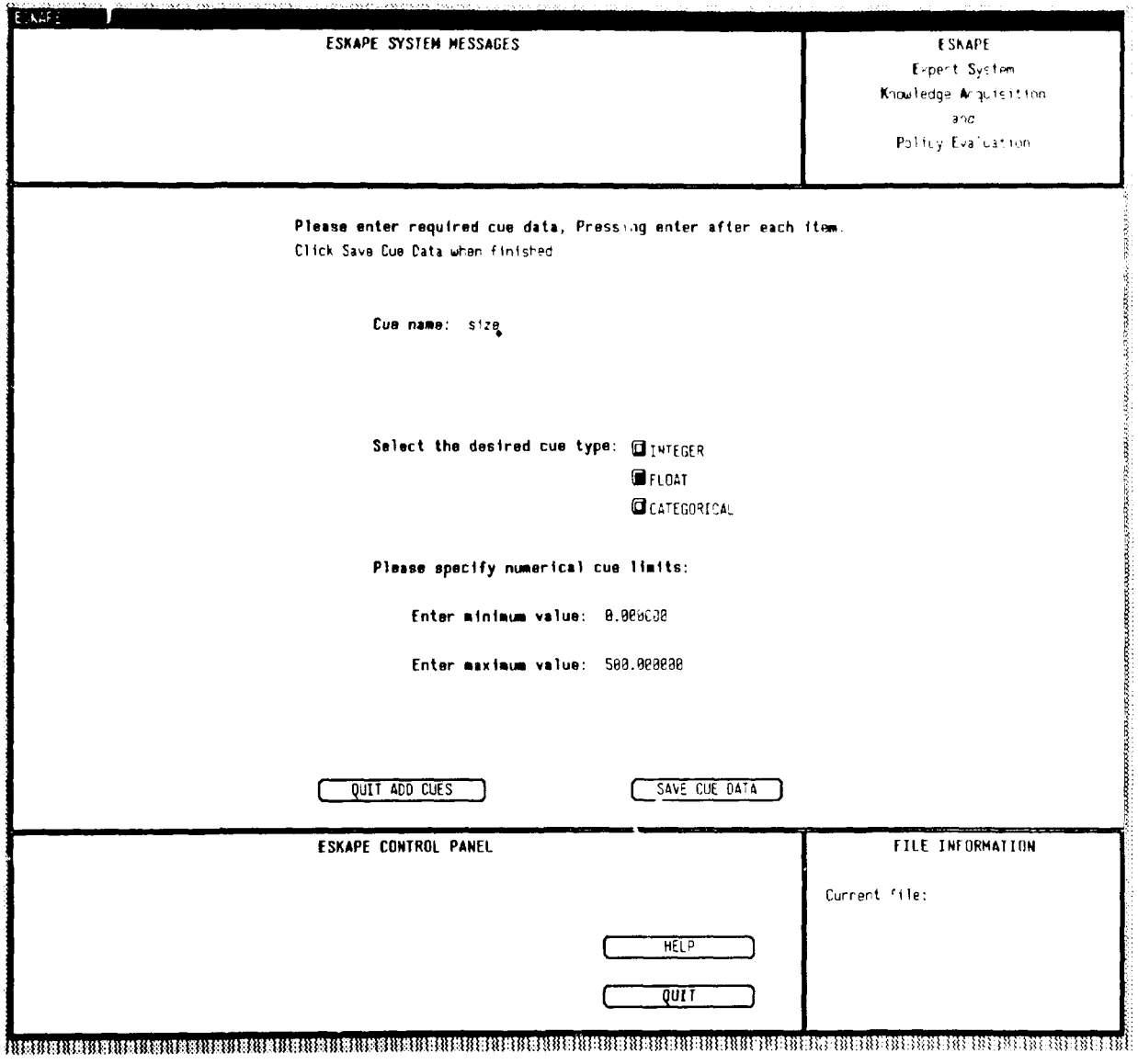

Figure 6. Adding first decision cue.

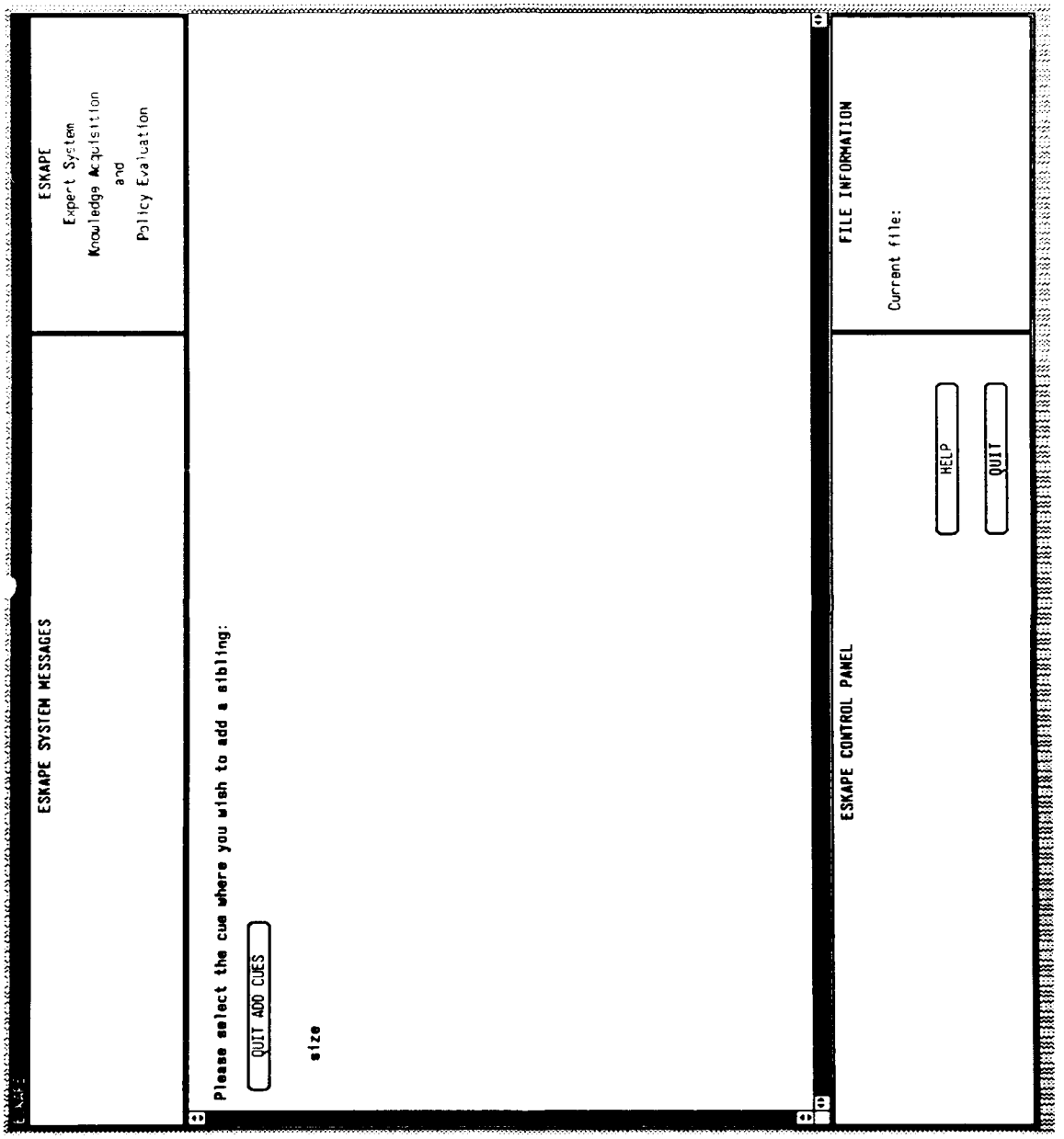

Figure 7. Edit selection screen after adding cue "size."

example, the user selects *size* (the only choice) and another cue edit screen is presented. In this case the user enters the categorical decision cue *program attrib* which has values ranging from "very low" (VL) to "very high" (VH) as shown in Figure 8. This cue addition process continues by adding two other decision cues, *person attrib* and *project attrib*.

4. Exploding Cues

After the first tier of decision cues has been created, the expert can explode them into further subcues as required. After selecting the EXPLODE CUE button on the ESKAPE control panel, the selection screen in Figure 9 is displayed. The user then selects the *size* cue to explode and enters appropriate data for the *KSLOC* cue on the forthcoming edit screen. After saving the *KSLOC* cue, the selection screen appears as in Figure 10. The other first tier cues are also exploded resulting in the selection screen in Figure 11.

At this point, further subcues can be entered by exploding the first tier cues or by adding new cues (ADD CUE) directly to the second tier, as shown in Figure 12. As levels of cues are entered, the screen scrolls so that all cues may not be visible for selection. In this case, the user can click on the appropriate scroll bar to display "hidden" cues.

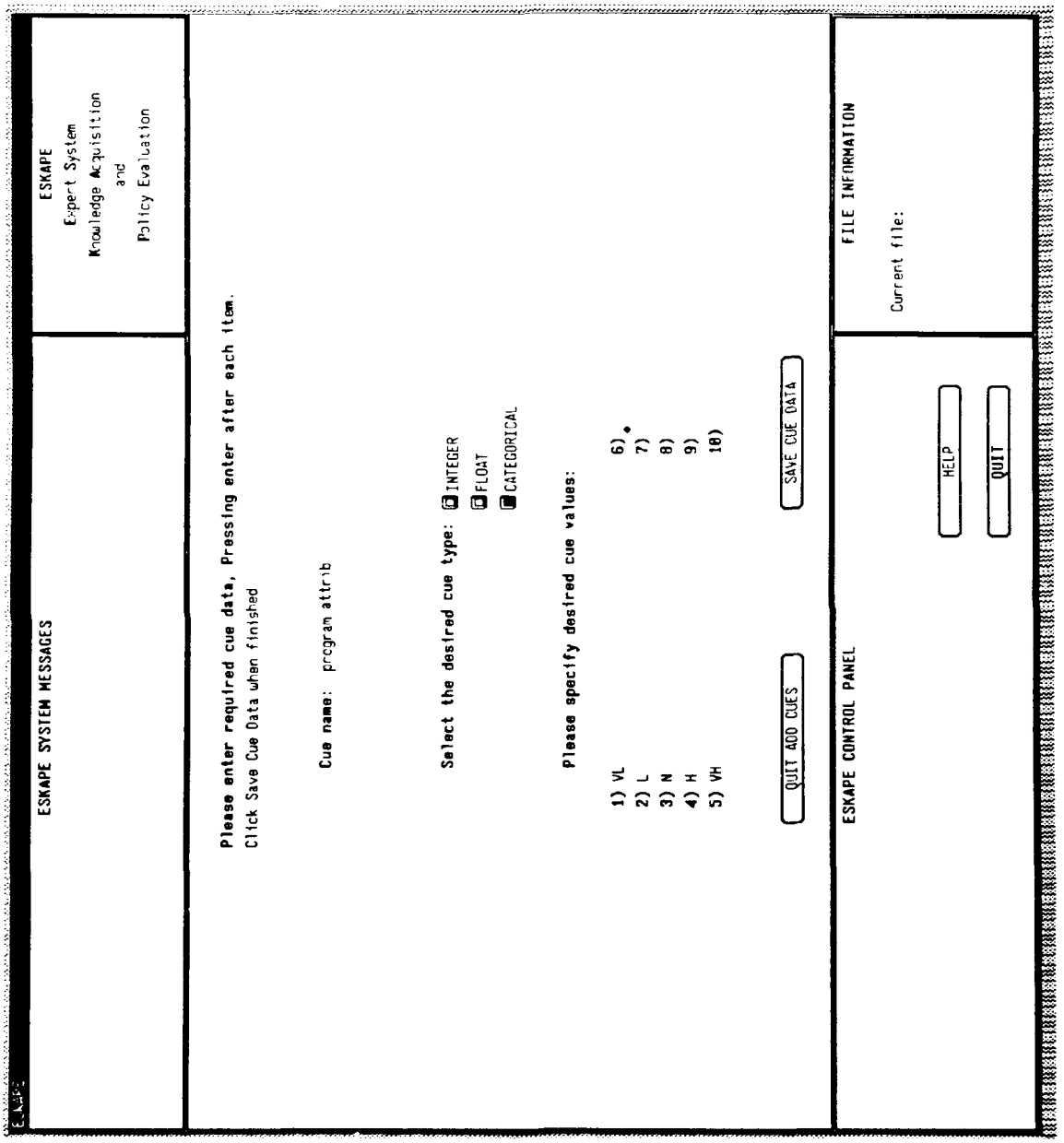

Figure 8. Adding cue "program attrib."

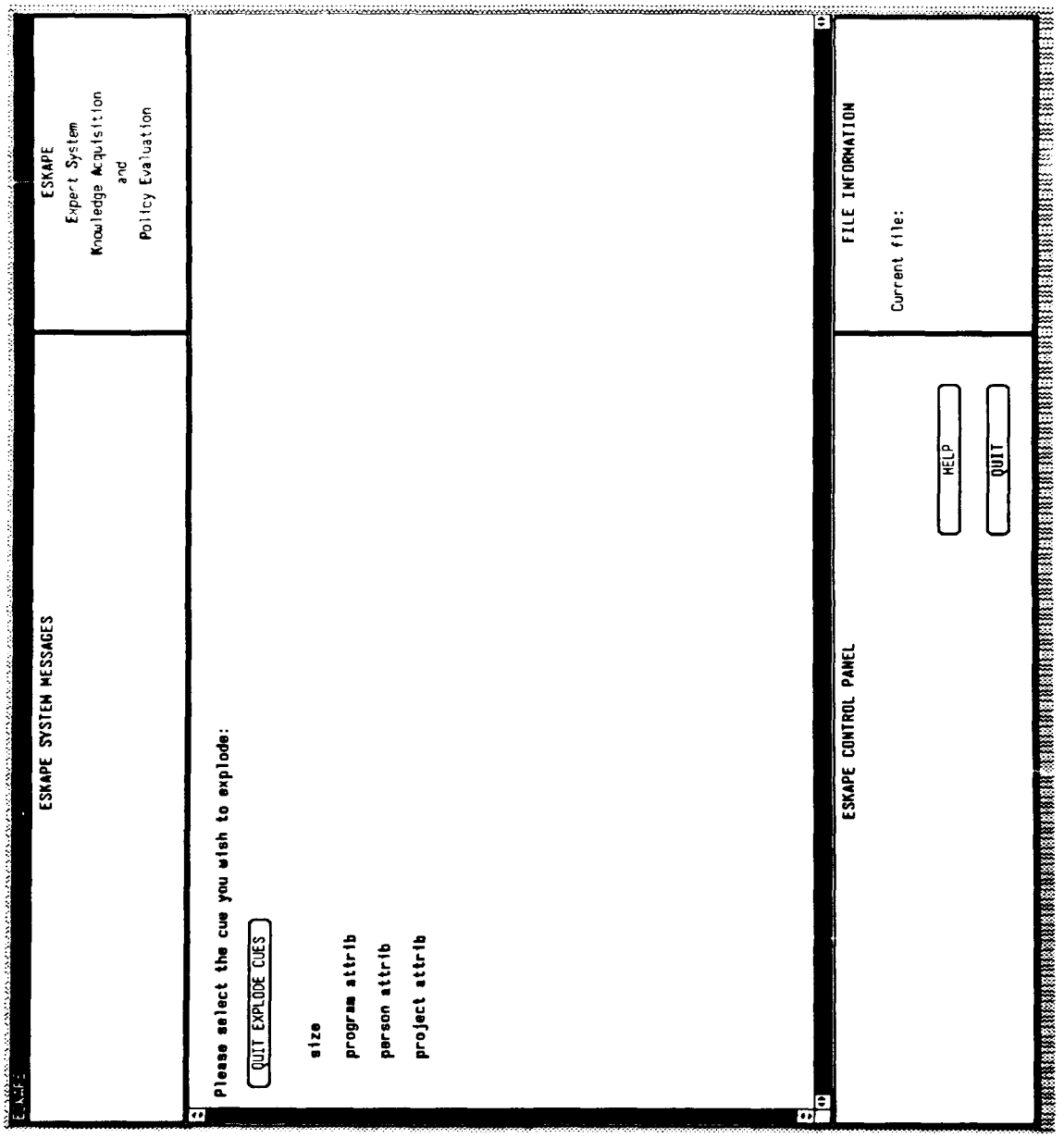

Figure 9. Explode cue selection page.

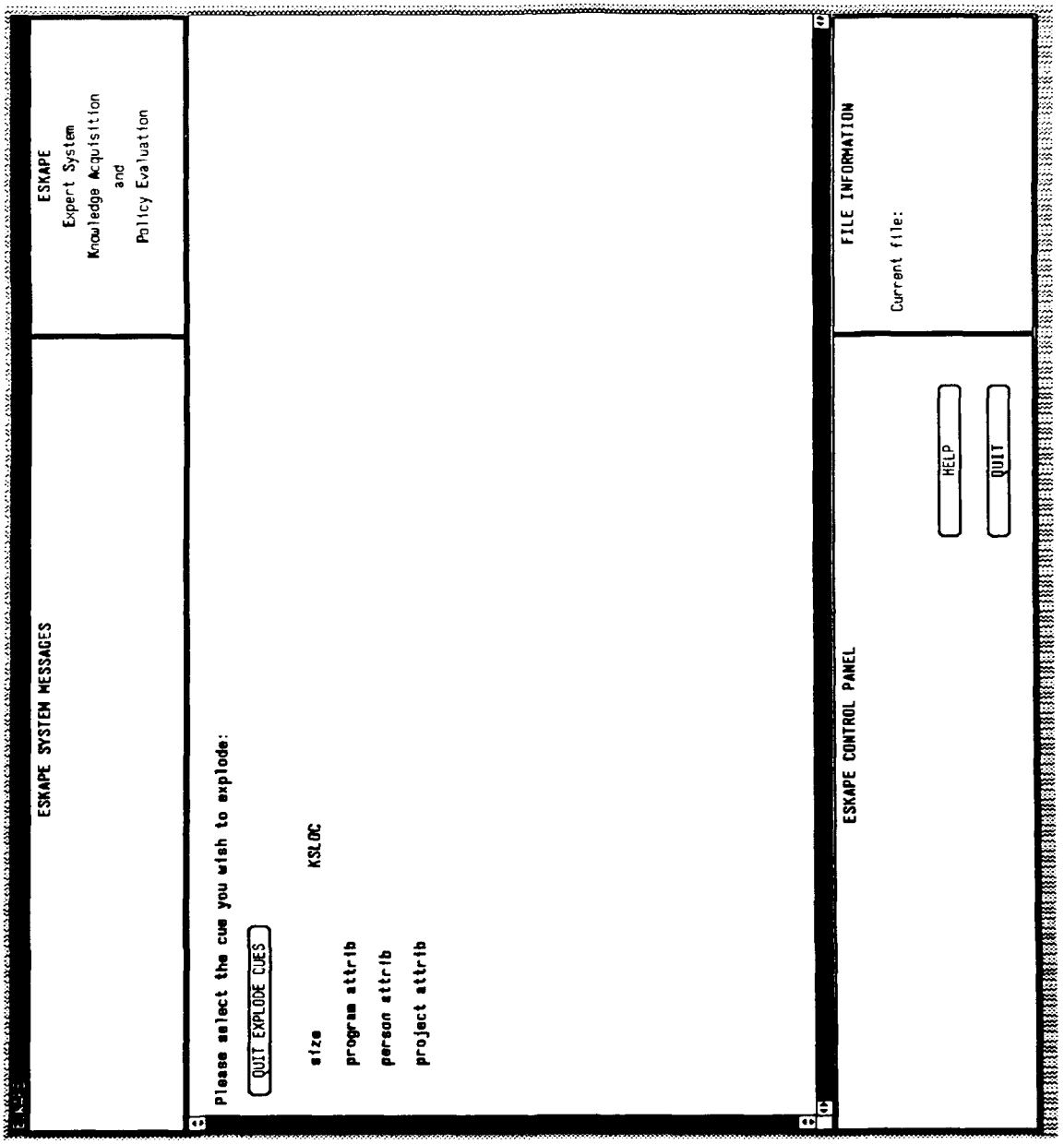

Figure 10. Explode cue selection page after "KSLOC."

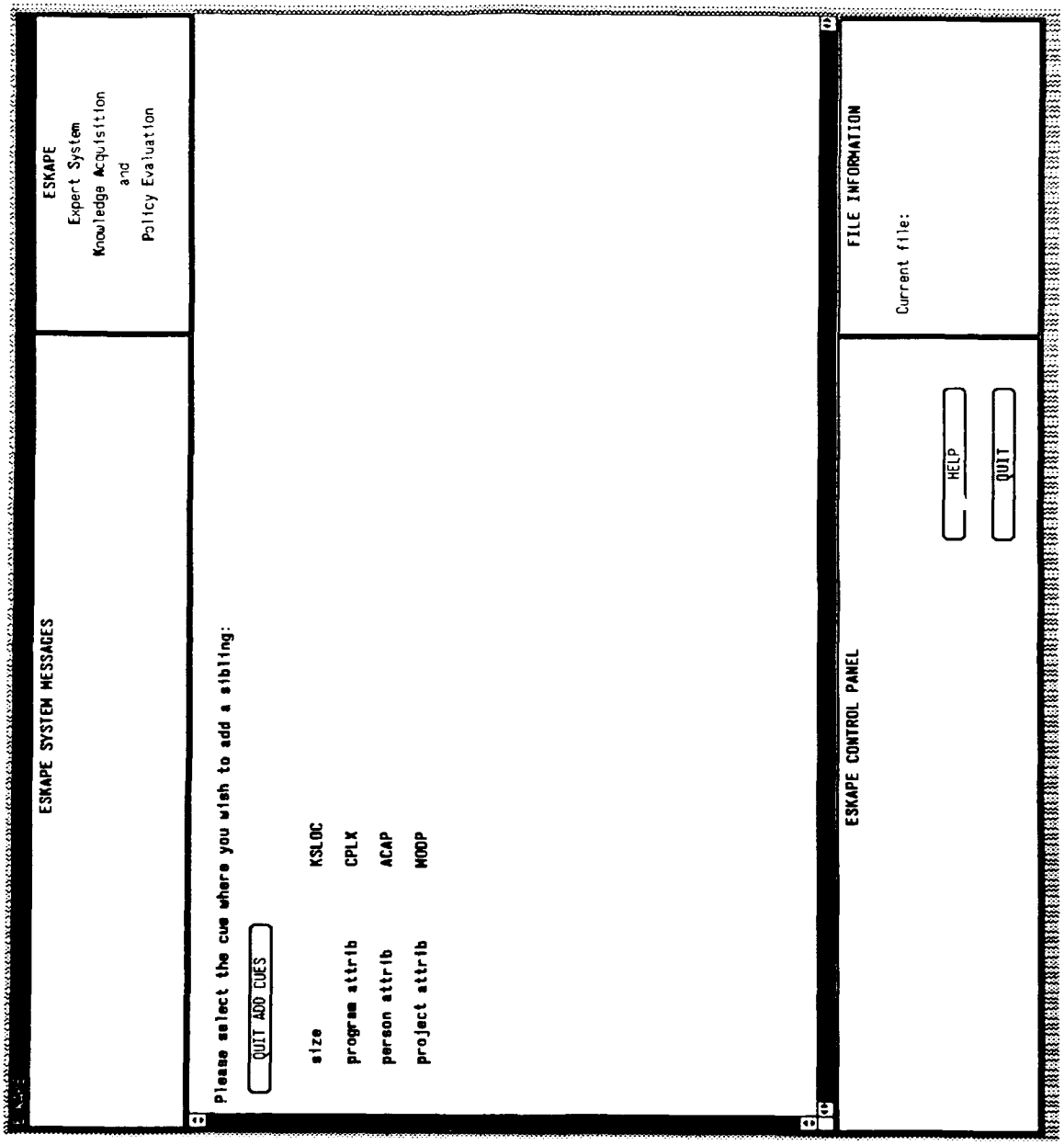

Figure 11. Explode cue selection page after exploding all.

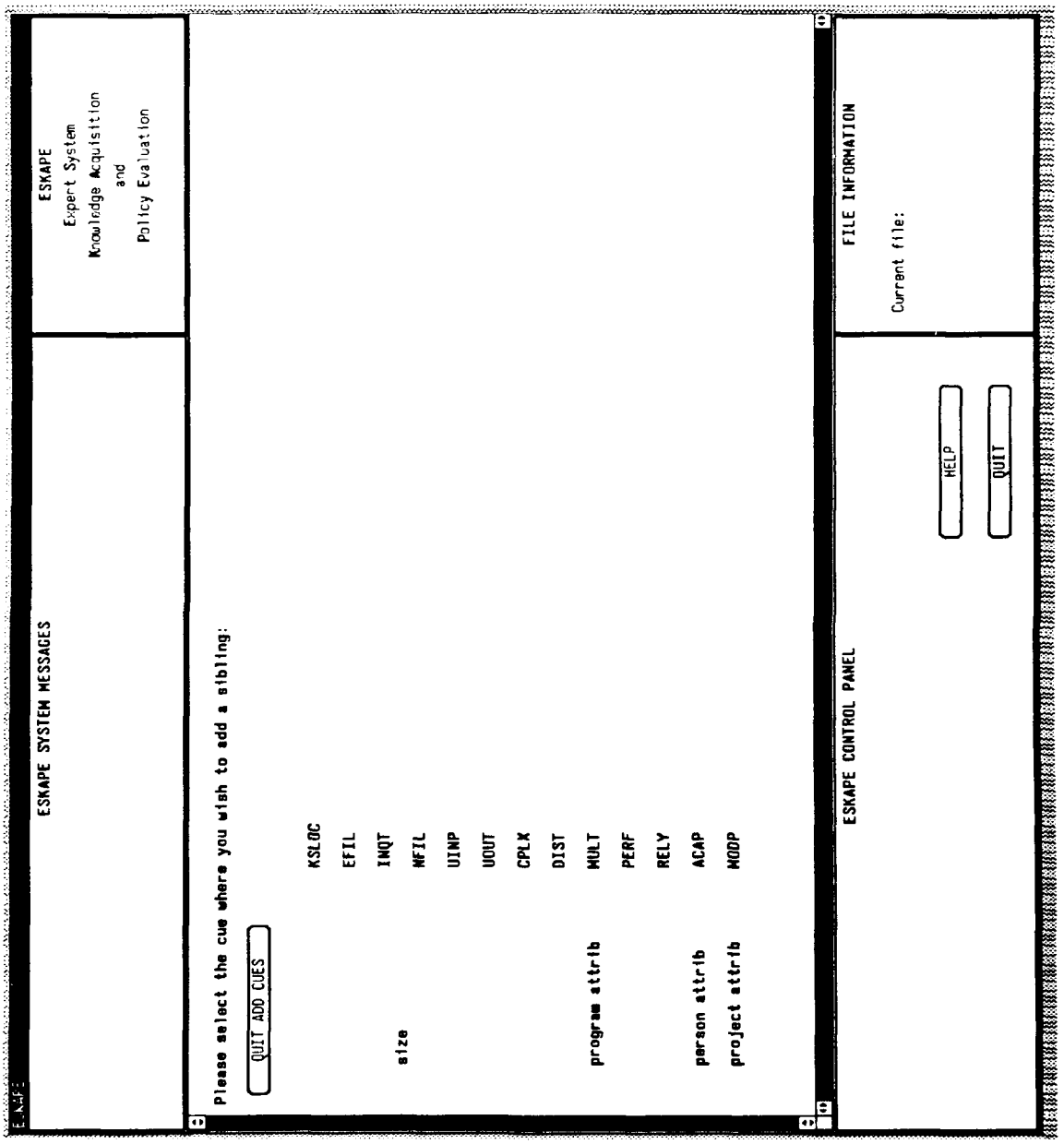

Figure 12. Add cue selection page after adding all siblings.

5. Correlating Cues

After entering the decision cues, the expert enters his or her estimates of the correlation between combinations of cues. To correlate cues, the user should first select CORRELATE CUES on the ESKAPE control panel, yielding a selection screen similar to Figure 13. The user then selects the first of the two cues he or she wishes to correlate[1]. Once selected, the first cue flashes and the system waits for the user to select a sibling. The user can "deselect" the first cue by clicking on that cue when it is visible.

When the second cue to be correlated is selected, a correlation screen similar to Figure 14 is displayed. The user then points to the desired correlation factor in the slider bar and presses the left mouse button.

The user is allowed to correlate decision cues because many naturalistic systems can be described by correlated cues (e.g., height/weight, smoking/cancer, etc.). The system limits this correlation factor to ±40% because Monte Carlo simulation revealed that ±40% was the largest generable correlation for the random data sets used by the program. Additionally, highly correlated cues introduce data redundancy and create a multicollinearity problem, degrading the accuracy of the regression model (Fienberg 1977). If two cues are

---

[1]The SAVE FILE button is now visible because the knowledge base was saved after all the cues were entered. See section B.13 for further information on saving the knowledge base.

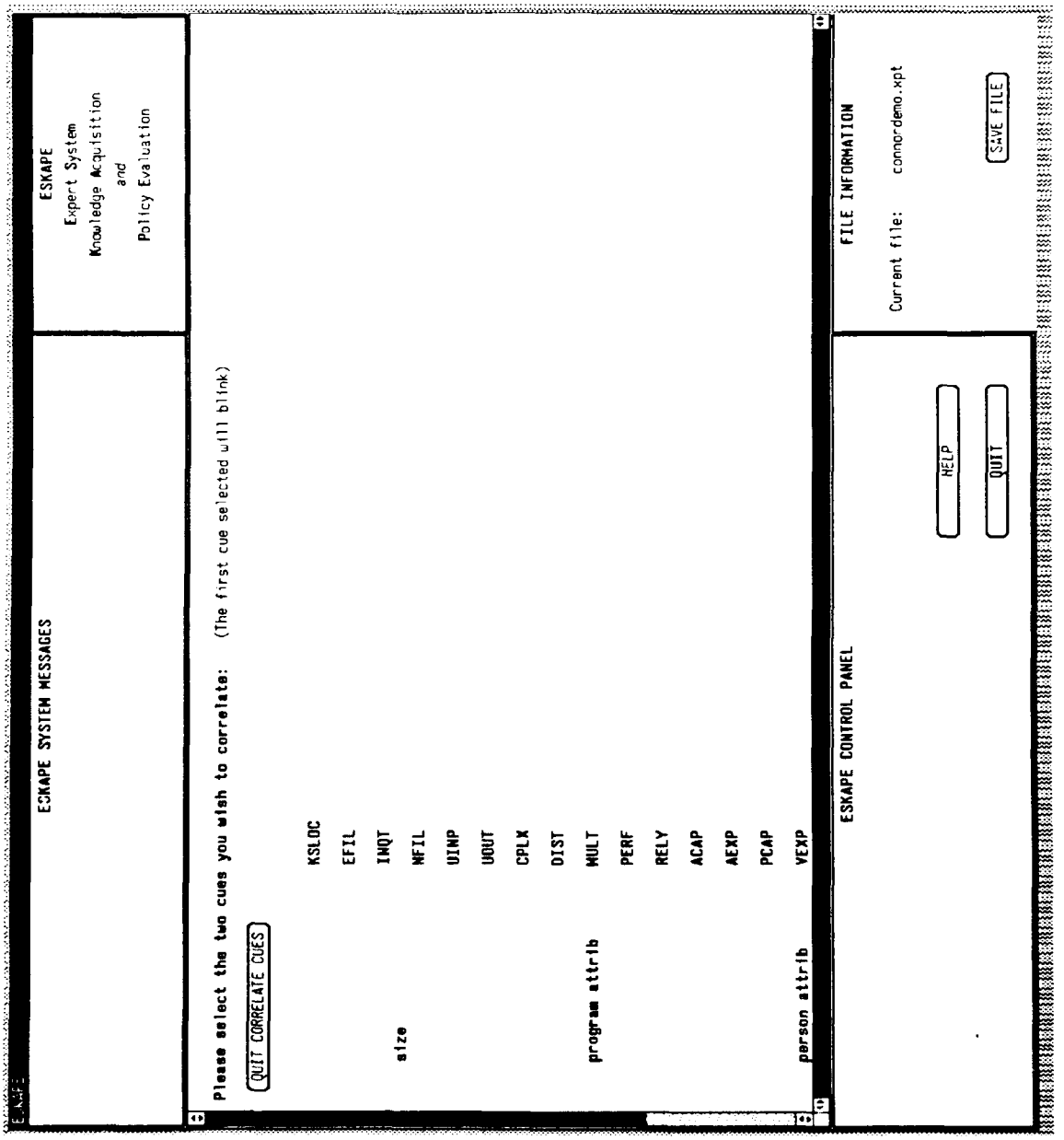

Figure 13. Correlate cue selection page.

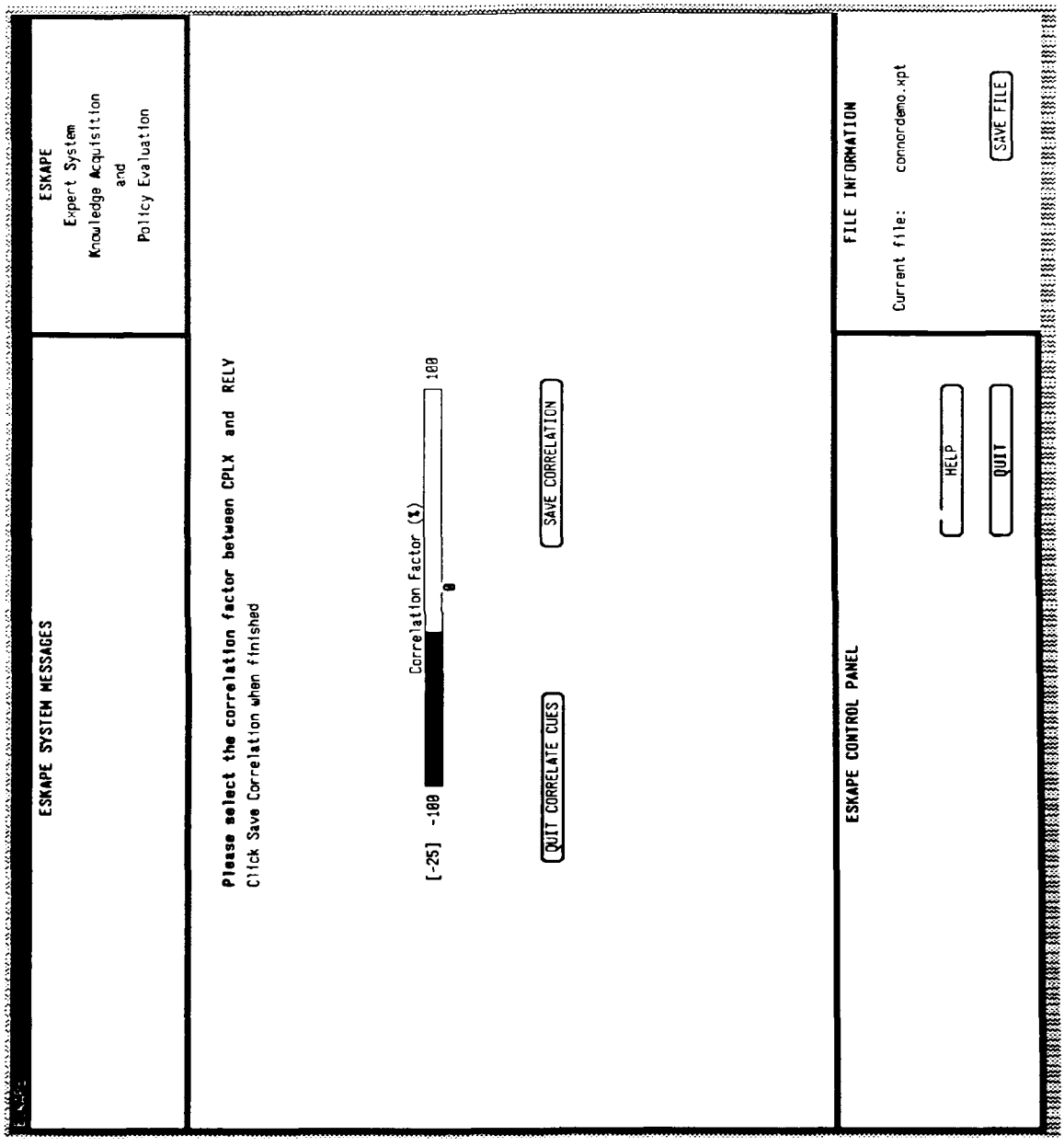

Figure 14. Setting the correlation factor.

highly correlated, the user should only use one cue and delete the other. If the user is unsure regarding the correlation between a pair of cues, he or she should leave the correlation set at zero.

6. Evaluating Cues

Once the cues have been created and correlated, the user can evaluate decision cues with two or more children. (If a cue has only one child, the cue and the child are the same and the child should be deleted.) The user selects EVALUATE CUE from the ESKAPE control panel and is presented with a selection panel similar to Figure 15. The user then selects the cue he or she wishes to evaluate and the program generates sample cases for the user to evaluate.

The program generates random cases within the limits and correlations specified by the user. The number of cases generated depends on the number of decision cues -- 30 cases for up to three cues, with an additional five cases for each cue above three. This ensures that sufficient cases are generated to provide stability in regression analysis. (Stewart 1988). The evaluation panel for the *program attrib* cue is shown in Figure 16. The user is asked to enter his or her judgment of the *program attrib* based on the values of the five decision cues -- *CPLX*, *DIST*, *MULT*, *PERF*, and *RELY*. The allowable values for the judgment node can be displayed at

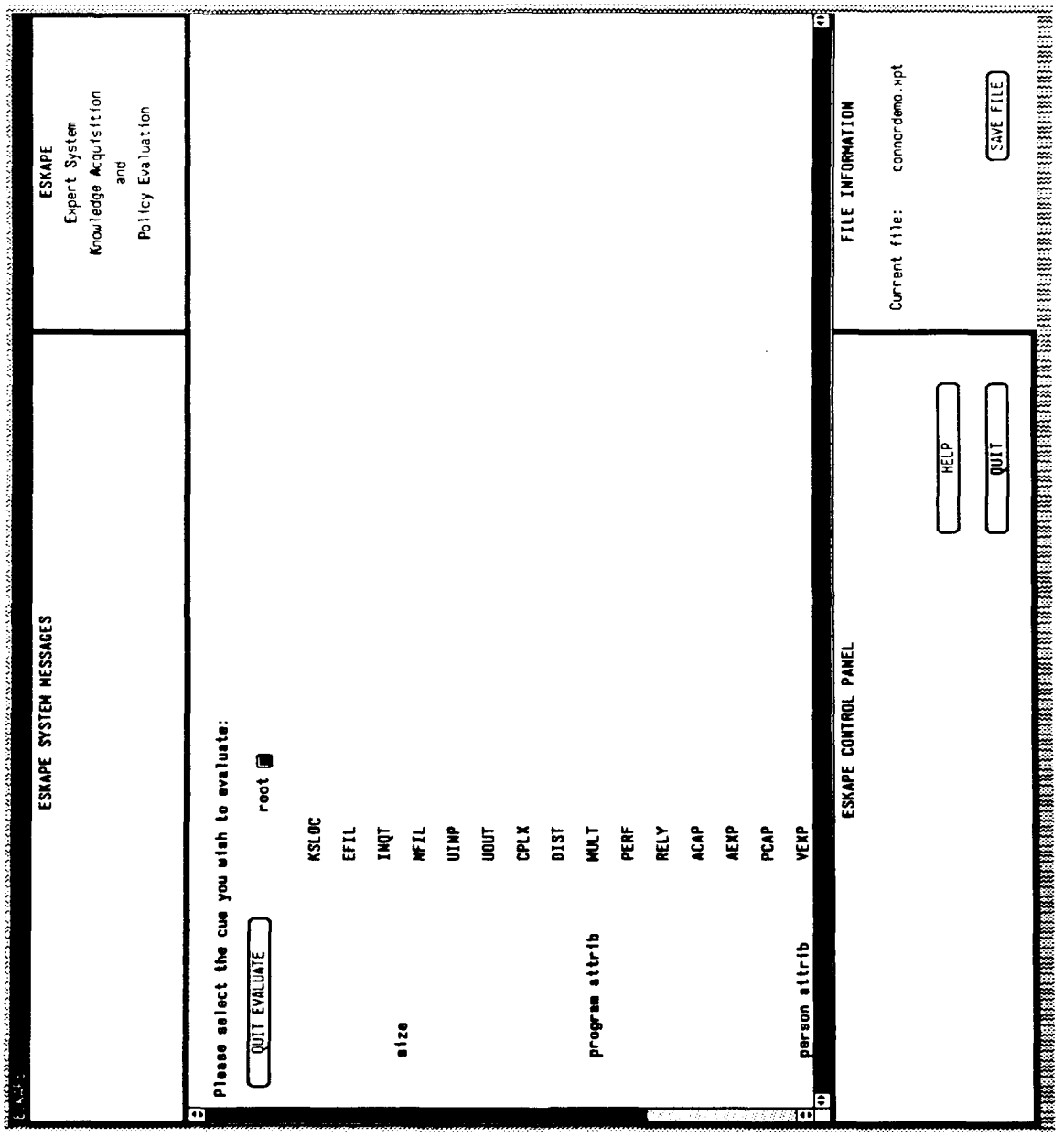

Figure 15. Evaluation cue selection page.

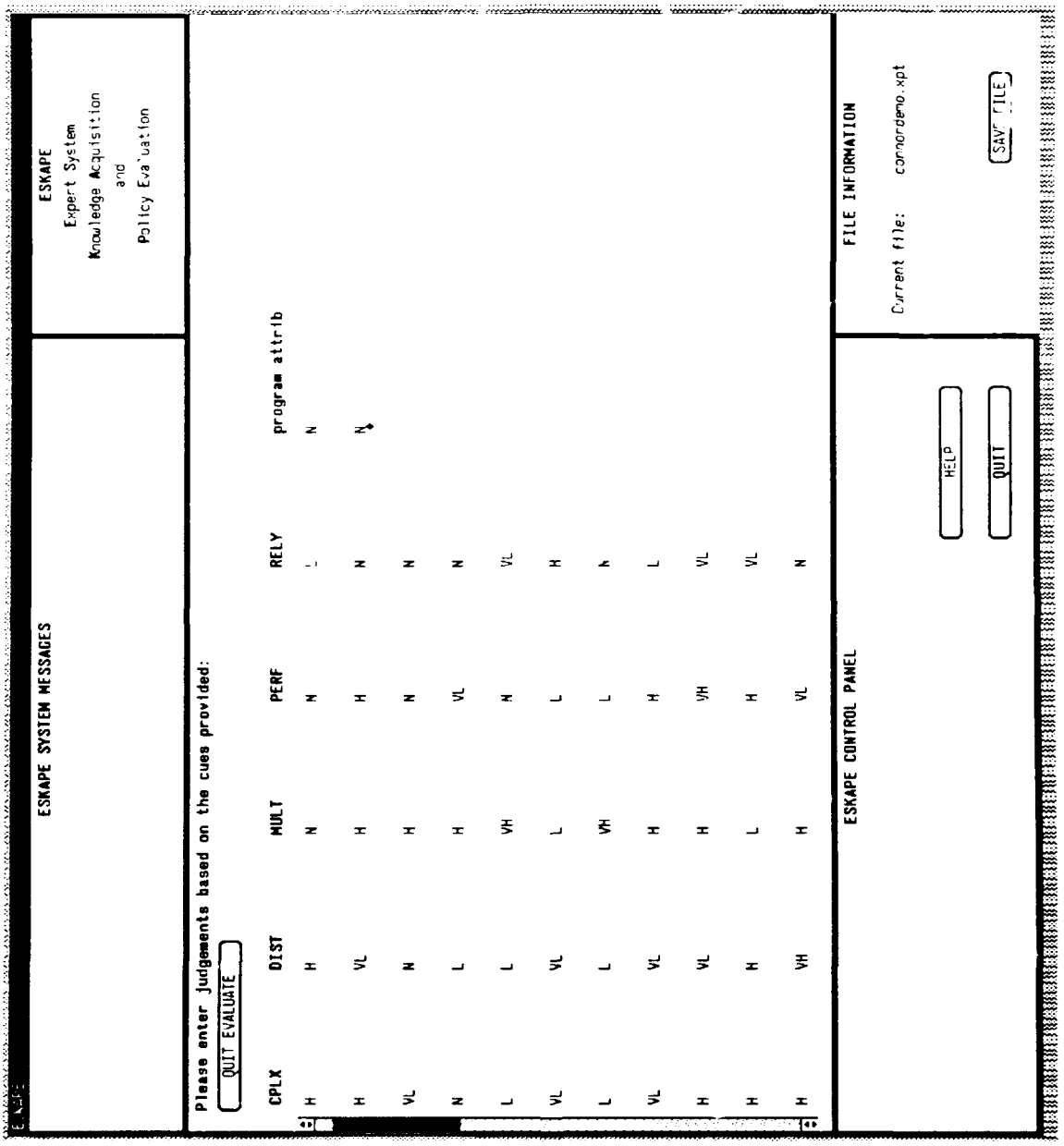

Figure 16. Evaluating cue "program attrib."

this point, as in Figure 17, by selecting HELP on the ESKAPE control panel.

Since the evaluation page is scrollable, the user can go back and change a judgment after he or she makes it. Once all the judgments are made, the program calculates appropriate beta weights and presents feedback to the user.

7. Feedback

Feedback is provided to the user in three forms as reported by Patterson (1990). The default presentation is Cue Relationships, but the user can selectively view any presentation. If the user decides that the evaluation was valid, he or she must remember to save the knowledge base. If the program is terminated without saving, cue evaluations (and other information) might be lost.

At this point the expert will see if any of the cues have beta weights near zero, indicating that the expert really didn't consider them in the decision making process. In this case, it may be appropriate to delete those cues and reevaluate to get a more accurate picture of the decision making process.

a. Cue Relationships with Decision Weights

The cue that was evaluated is displayed as the head of a subtree as in Figure 18. Its decision cues are also depicted along with their corresponding beta weights. The

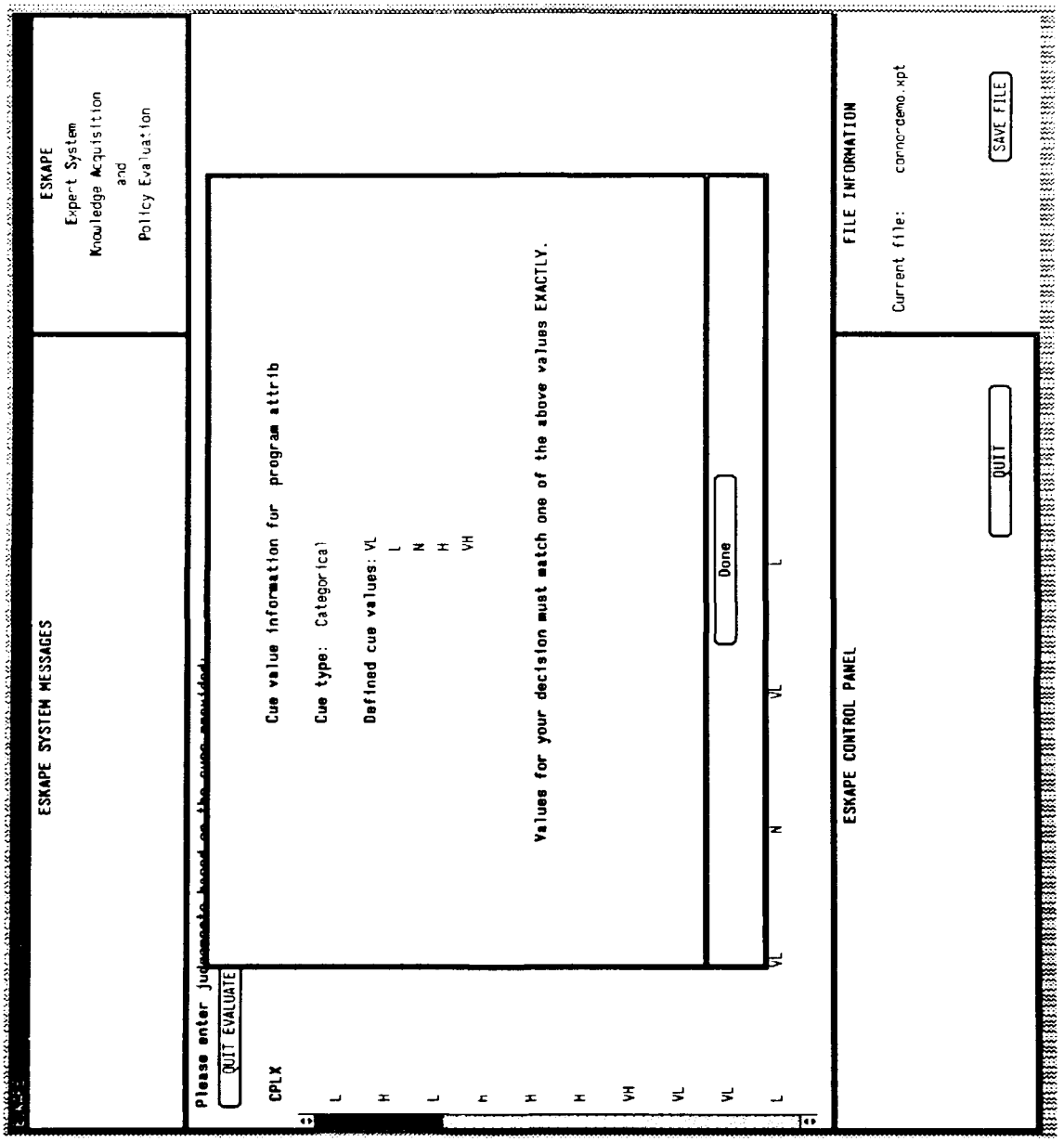

Figure 17. Evaluate cue help screen.

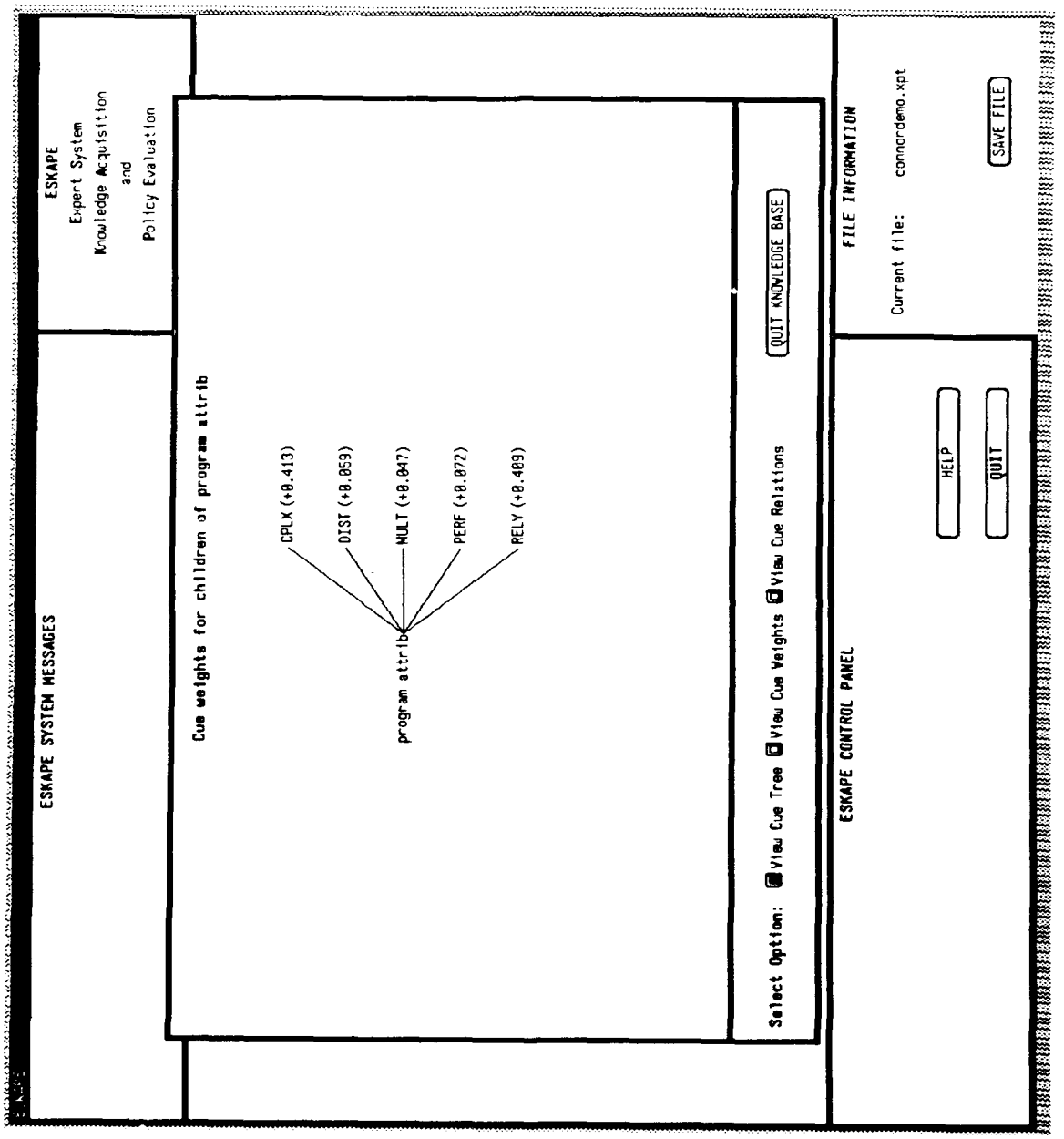

Figure 18. Cue relationship feedback.

beta weights indicate the degree to which the expert relies on a particular cue in the decision making process.

b. Decision Weights

The beta weights of for the decision cues are displayed in a stacked bar format, positive correlations above the axis, negative below. The beta weights indicate the degree to which the expert relies on a particular cue in the decision making process. This format is shown in Figure 19.

c. Function Forms

The values for each decision cue are plotted against the value of the judgment. This presentation, as in Figure 20, can show trends or relationships between individual cues and the judgment. For example, the slight upward trend shown in Figure 20 for cue *CPLX* may indicate a slight positive correlation between *CPLX* and *program attrib*. Similarly, the relatively flat relationship between *MULT* and *program attrib* may indicate that *MULT* has little or no bearing on the decision process.

8. Viewing the Knowledge Base

At any time, the expert can view the accumulated knowledge base by selecting KNOWLEDGE BASE on the ESKAPE control panel. The user must then elect to view the entire tree or view a decision cue that has been already evaluated, as shown in Figure 21. If the single cue option is chosen, a selection panel similar to Figure 22 is presented. After a

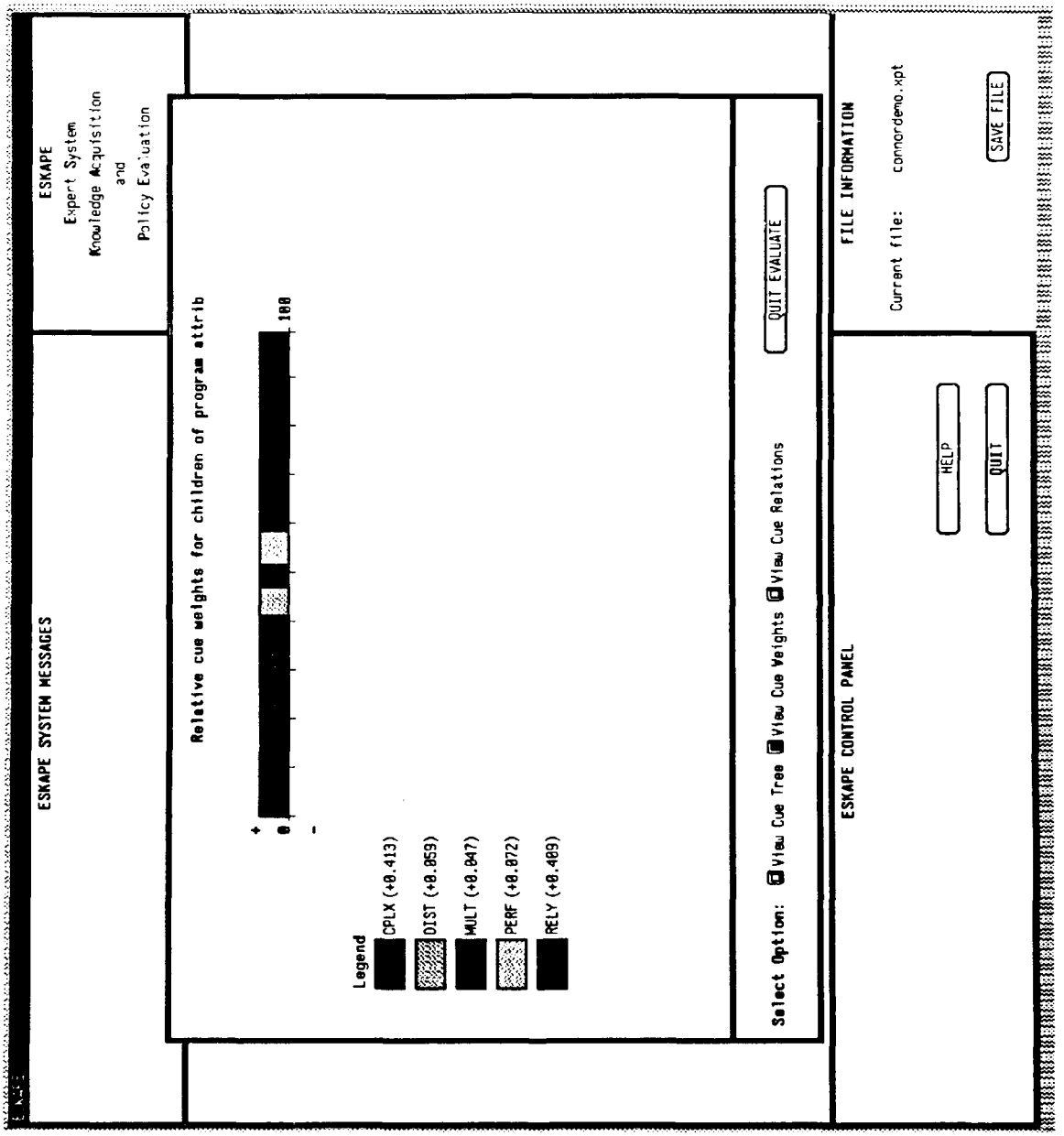

Figure 19. Decision weight feedback.

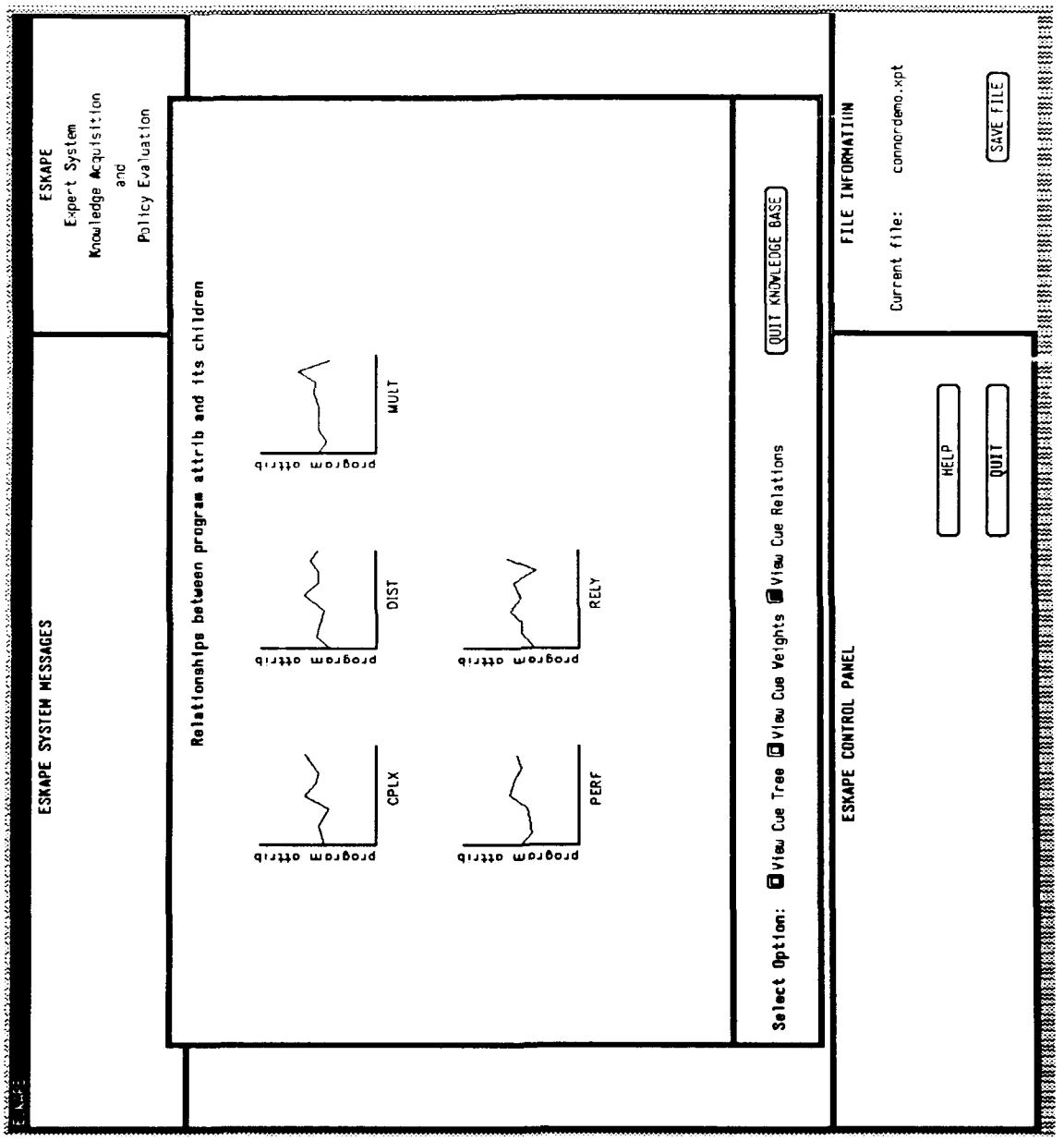

Figure 20. Function forms feedback.

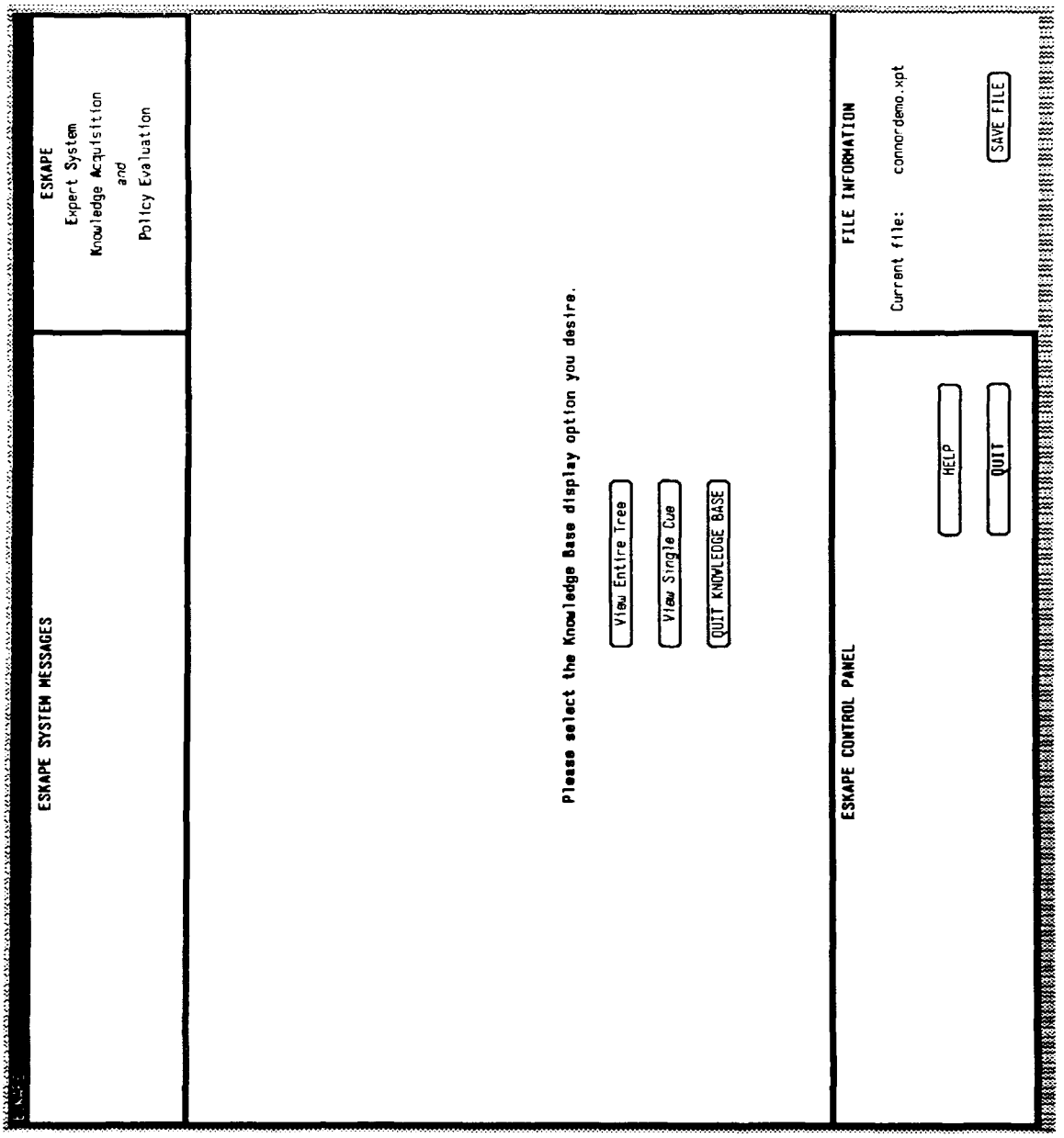

Figure 21. Knowledge base option selection page.

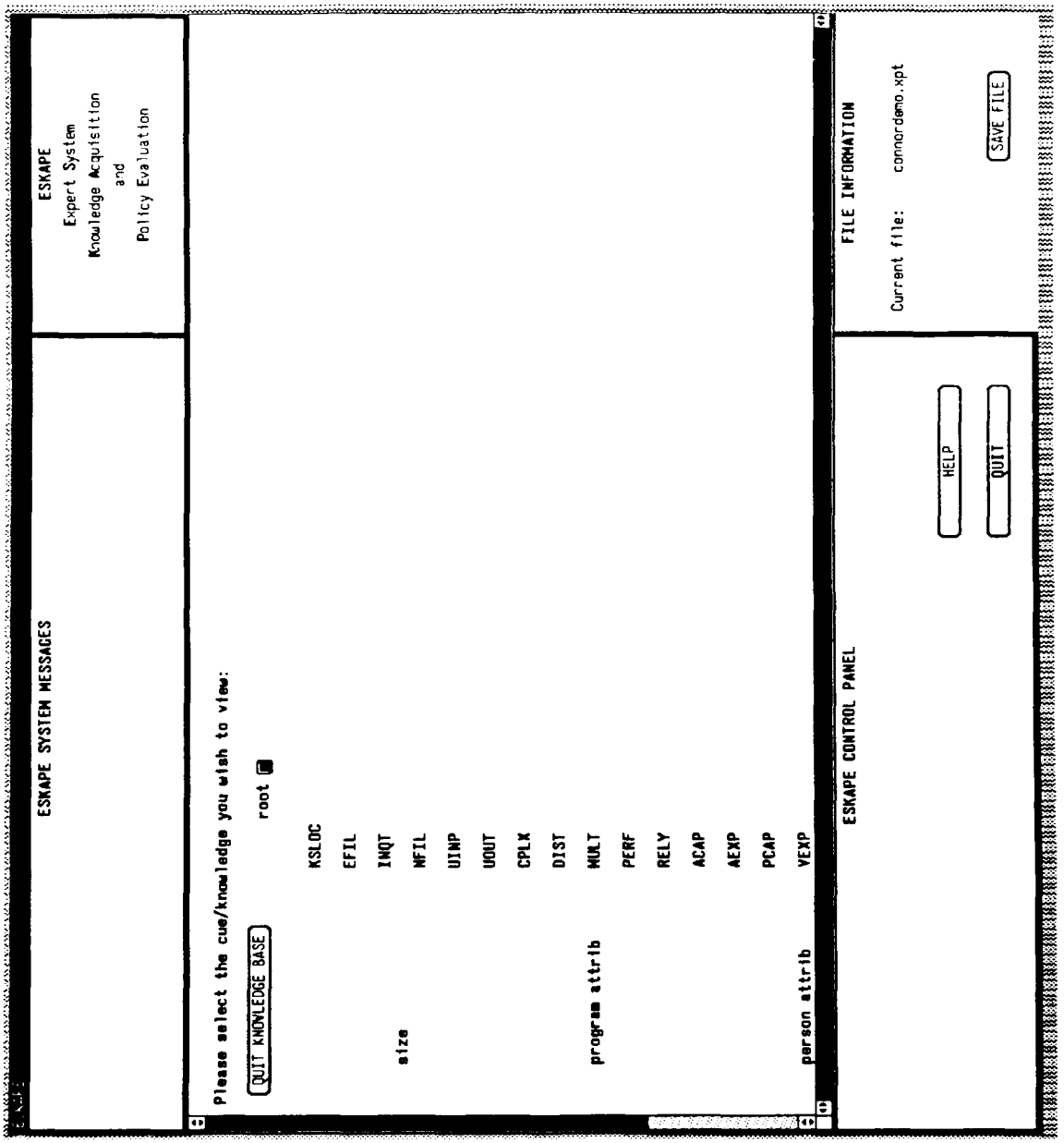

Figure 22. Singe cue knowledge base selection page.

specific cue is selected, the user will be shown the same screens generated during the evaluation of that cue.

If the user elects to view the entire knowledge tree, all decision cues, their children, grandchildren, etc. are displayed in a scrollable tree format. Additionally, as seen in Figure 23, the beta weights for evaluated cues are also displayed.

9. Generating Sample Cases

The expert can generate sample cases to validate his or her policies by selecting GENERATE CASES from the ESKAPE control panel. After selecting the appropriate cue from a selection screen, the expert is presented with numerous sample cases along with a judgment in each case based on the calculated decision weights, as in Figure 24. After viewing these judgments, the expert can either accept the outcomes or

1. Reevaluate the cue.
2. Edit one or more decision cues.
3. Delete one or more decision cues.
4. Correlate/recorrelate one or more pairs of decision cues.
5. Move decision cues around the knowledge tree.

The required knowledge has been captured if all cues with two or more children have been evaluated (including the root) and the expert is satisfied with the results of case generation in each case.

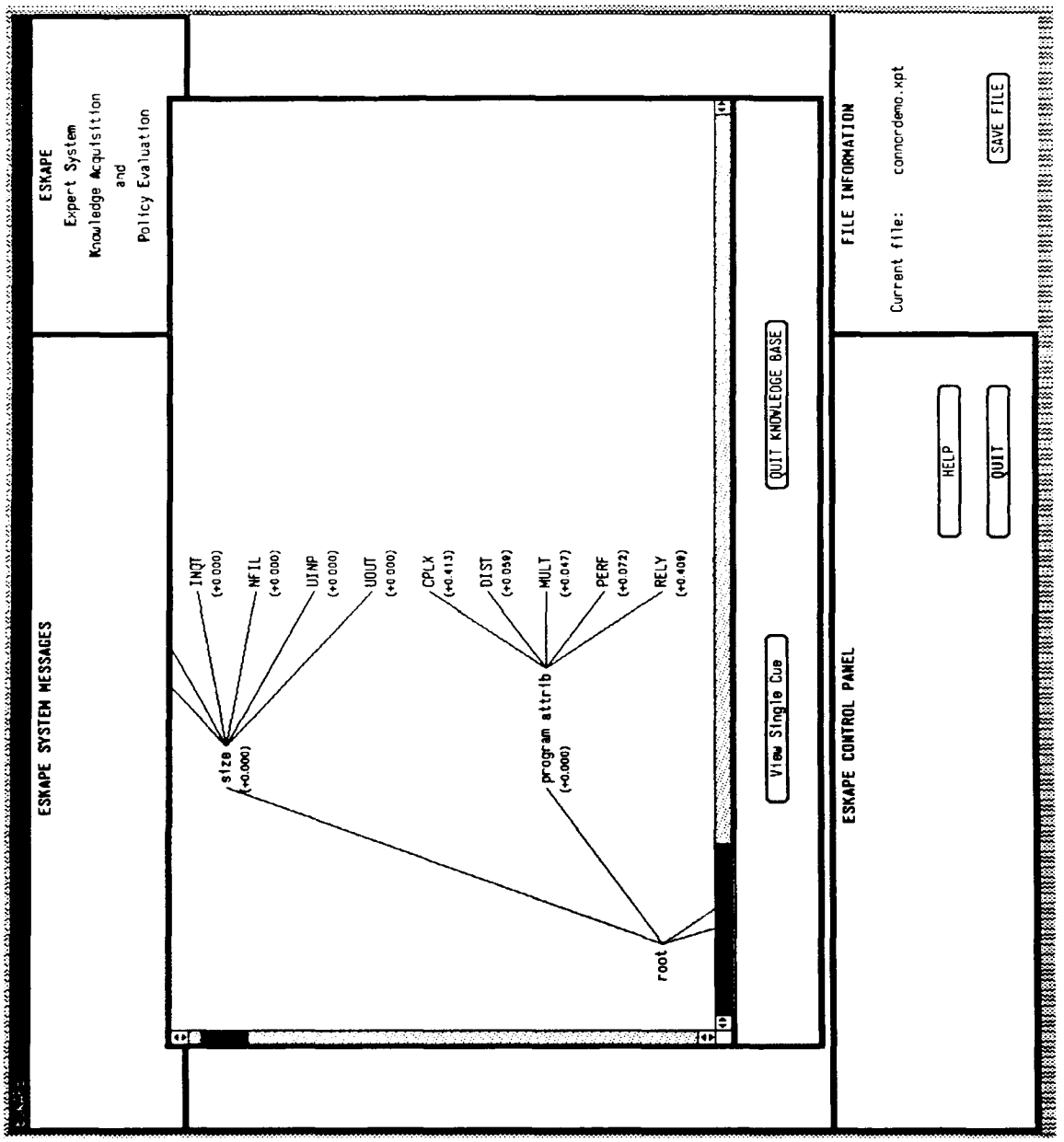

Figure 23. Entire knowledge tree.

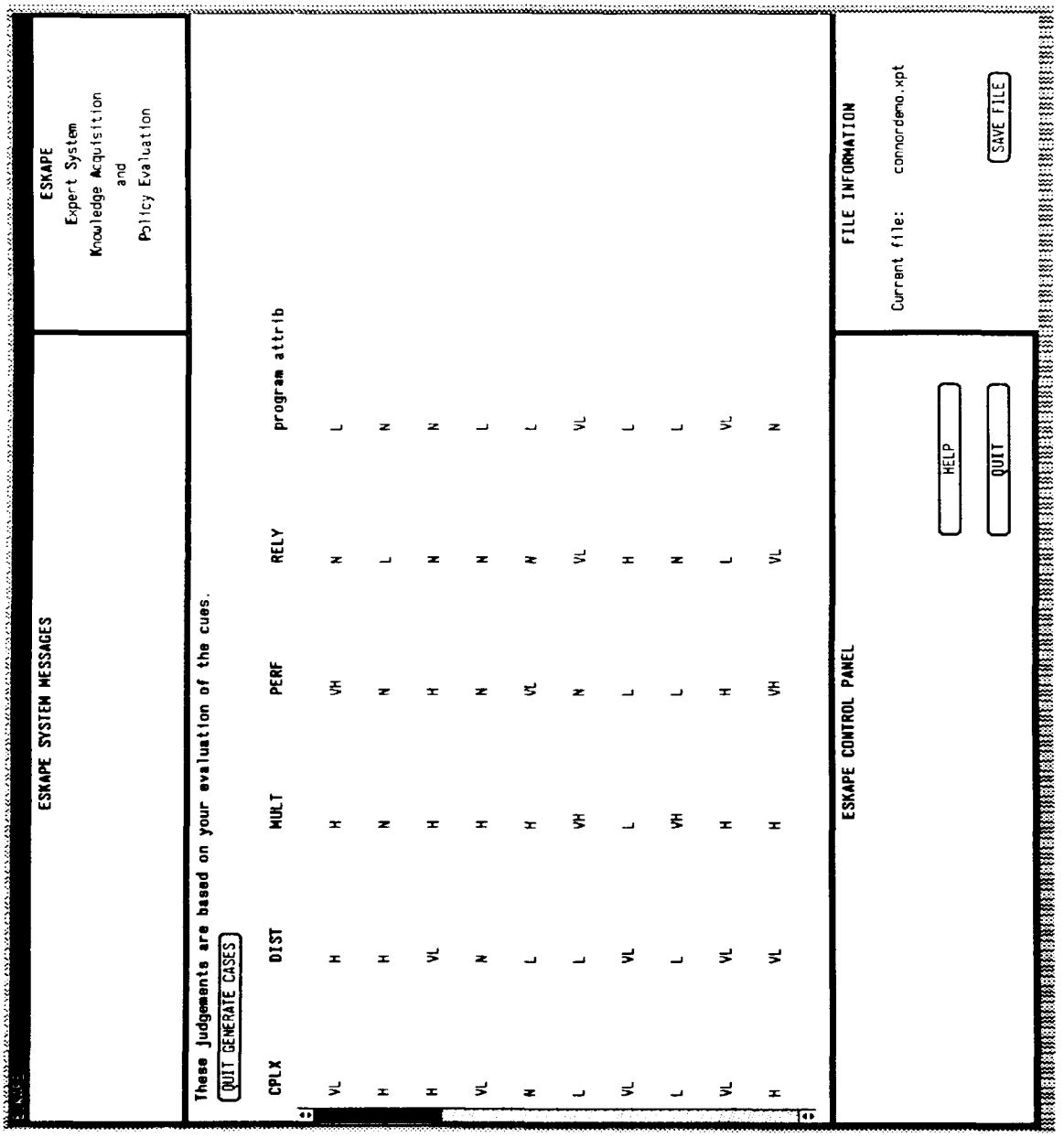

Figure 24. Generate cases screen for cue "program attrib."

10. Editing Cues

A cue may be edited once it has been created with either the add or explode cue functions. If EDIT CUE is selected on the ESKAPE control panel, a selection screen similar to Figure 25 is presented. An edit screen for the selected cue is then displayed which contains its previously entered values, as shown in Figure 26. The user then corrects the data as required and selects SAVE CUE DATA when finished. If a cue is edited, it must be reevaluated.

11. Deleting Cues

The user normally deletes a cue when he or she determines that the decision cue does not contribute to the decision making process. To delete a cue, the user selects DELETE CUE in the ESKAPE control panel, yielding the familiar selection screen of Figure 27. The user then selects and verifies that the cue should be deleted. After user confirmation, the cue and its entire subtree is deleted from the knowledge tree. Figure 28 shows the result of cue *size* being deleted.

12. Moving Cues

Moving a cue to a different part of the knowledge tree is a less destructive option than outright deletion. To move a cue, the user first selects MOVE CUE from the ESKAPE control panel. A selection screen similar to Figure 29 is

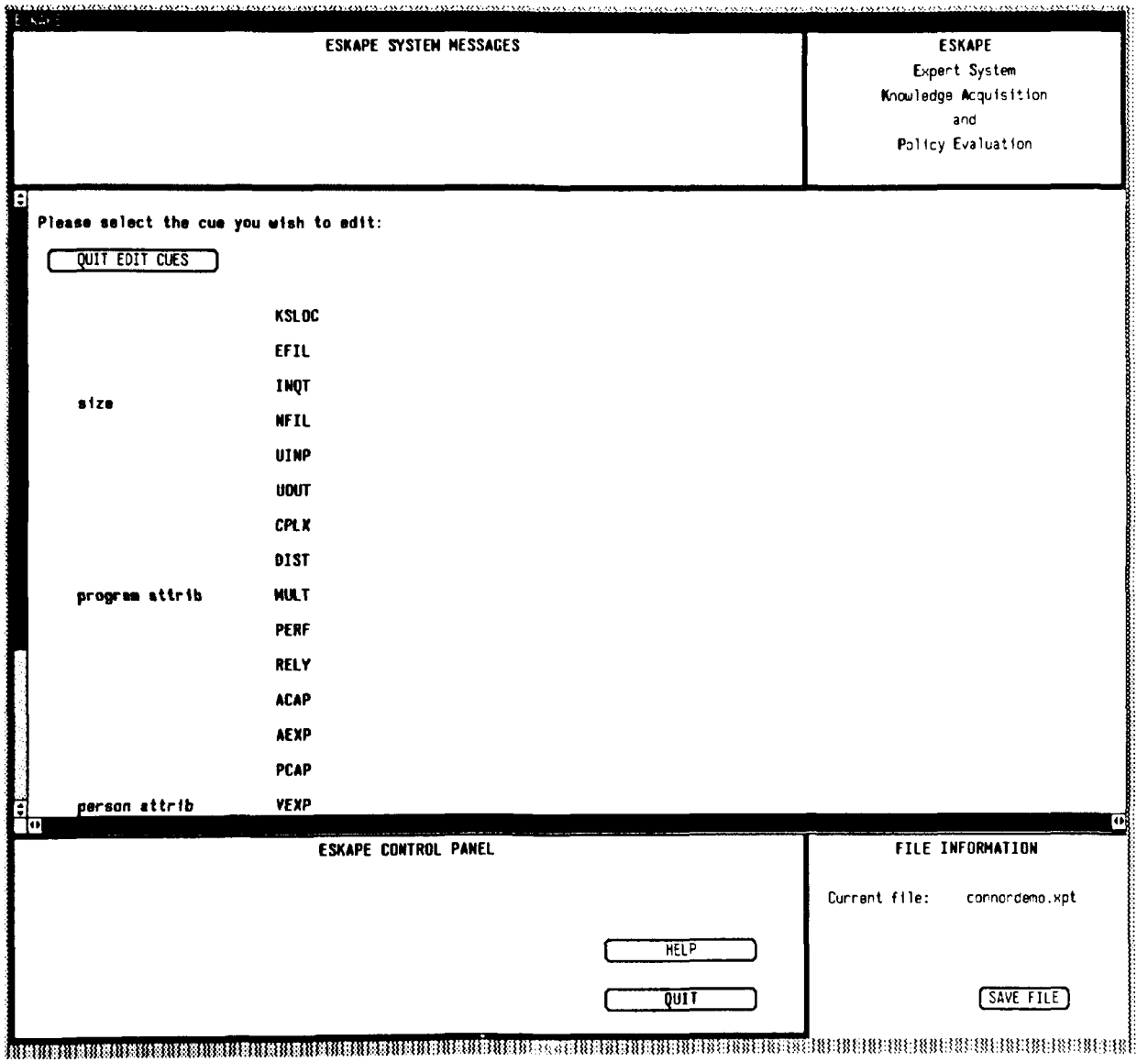

Figure 25. Edit cue selection page.

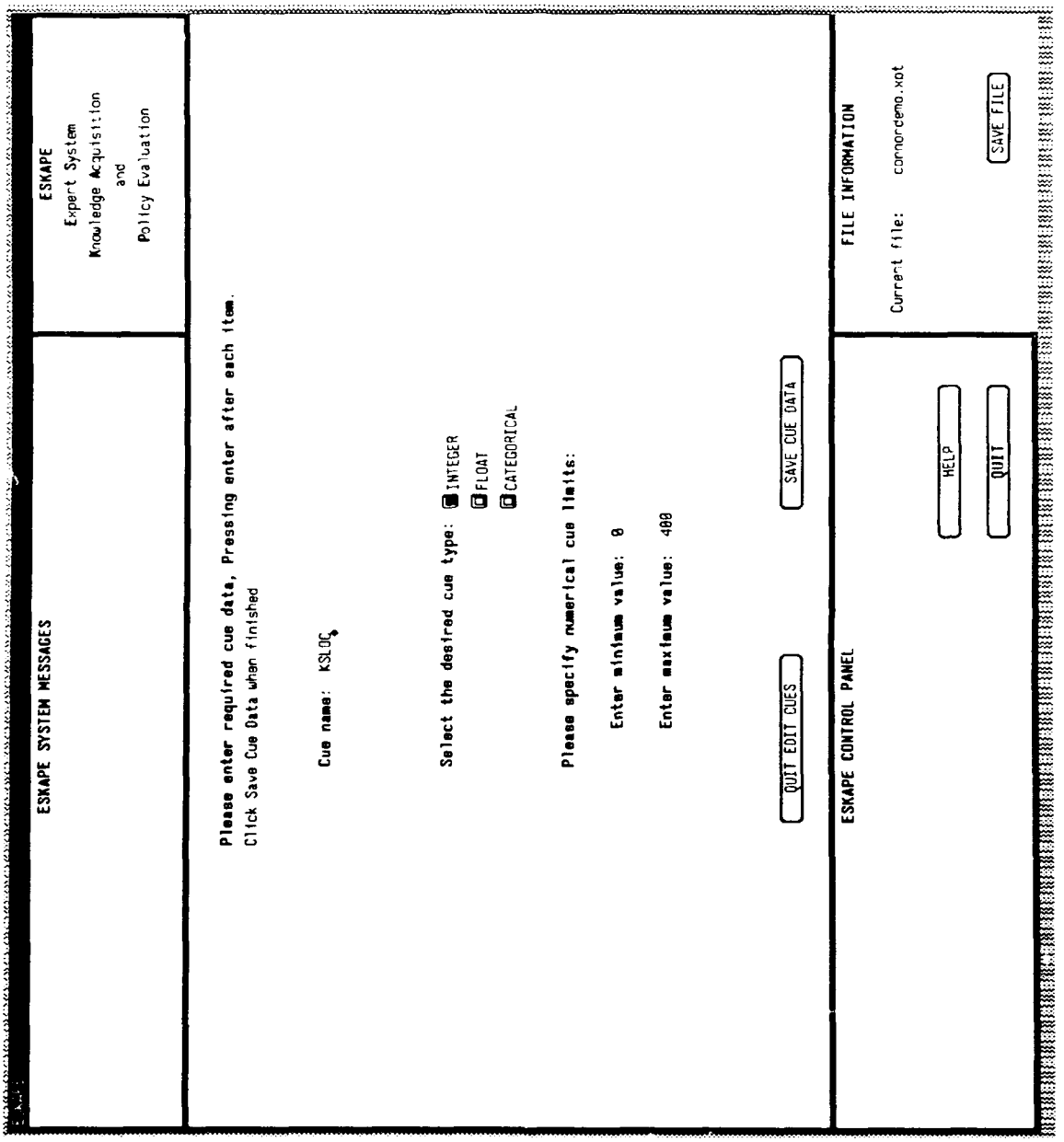

Figure 26. Edit page for cue "KSLOC."

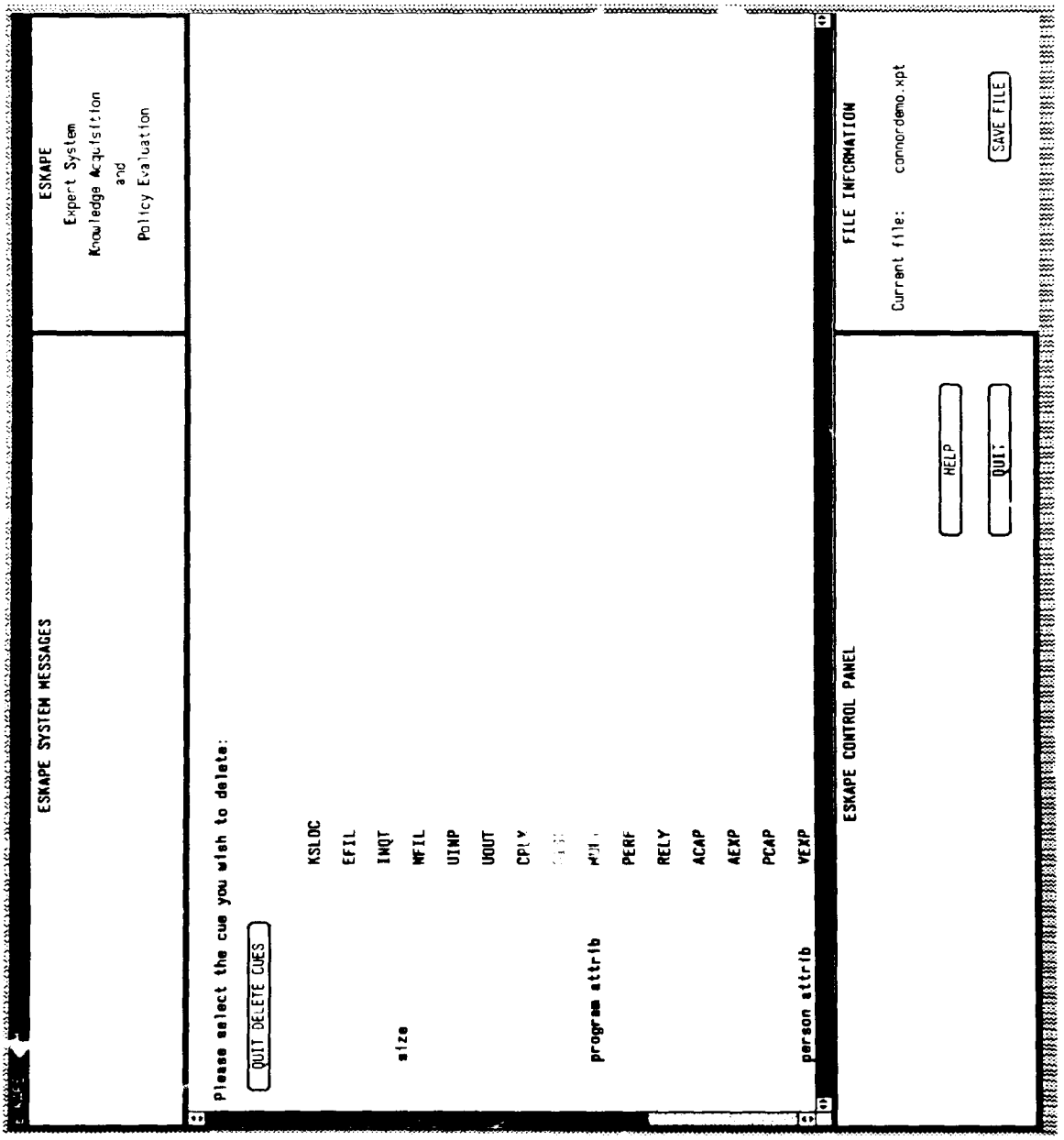

Figure 27. Delete cue selection page.

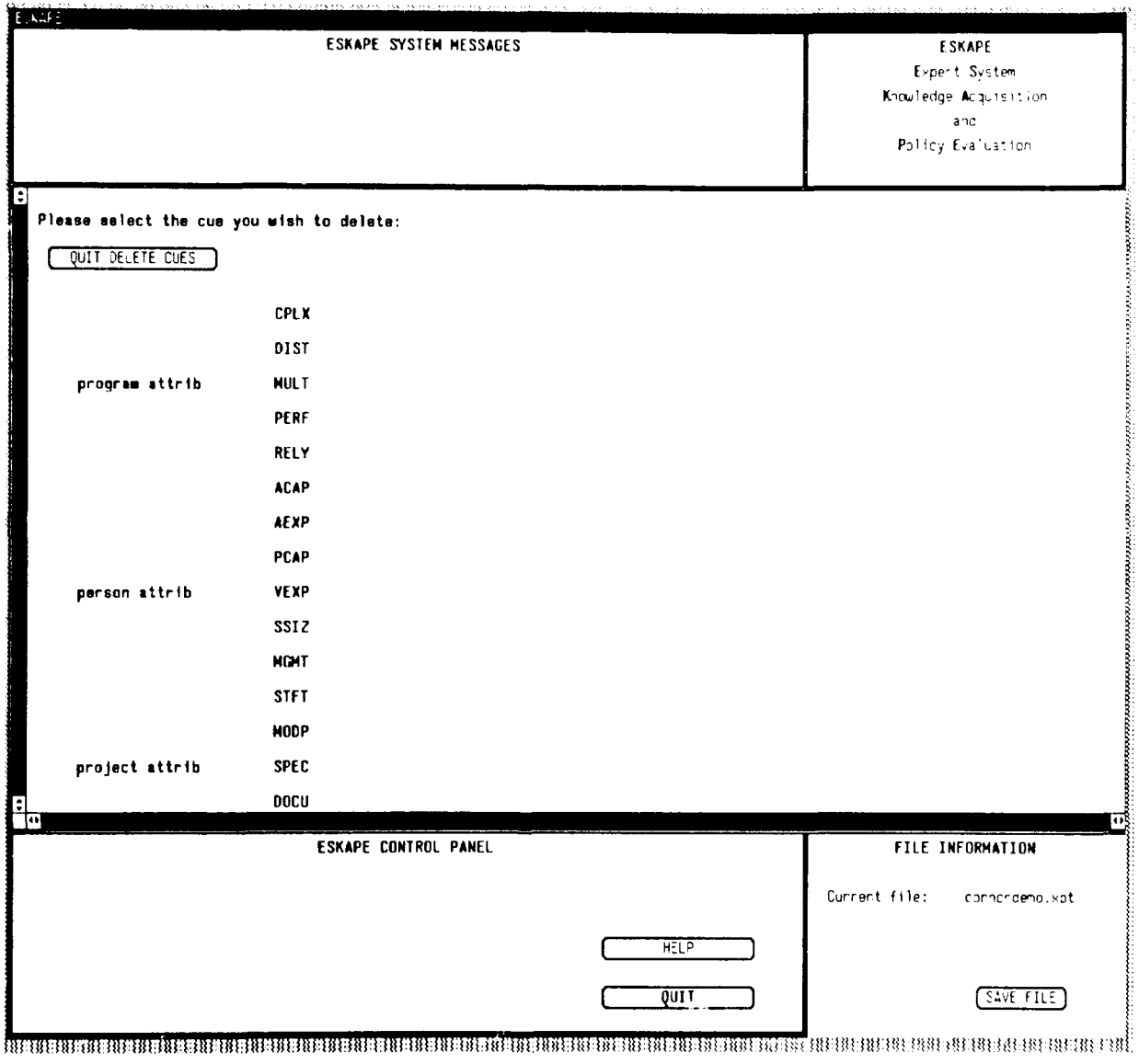

Figure 28. Delete cue selection page after deleting "size."

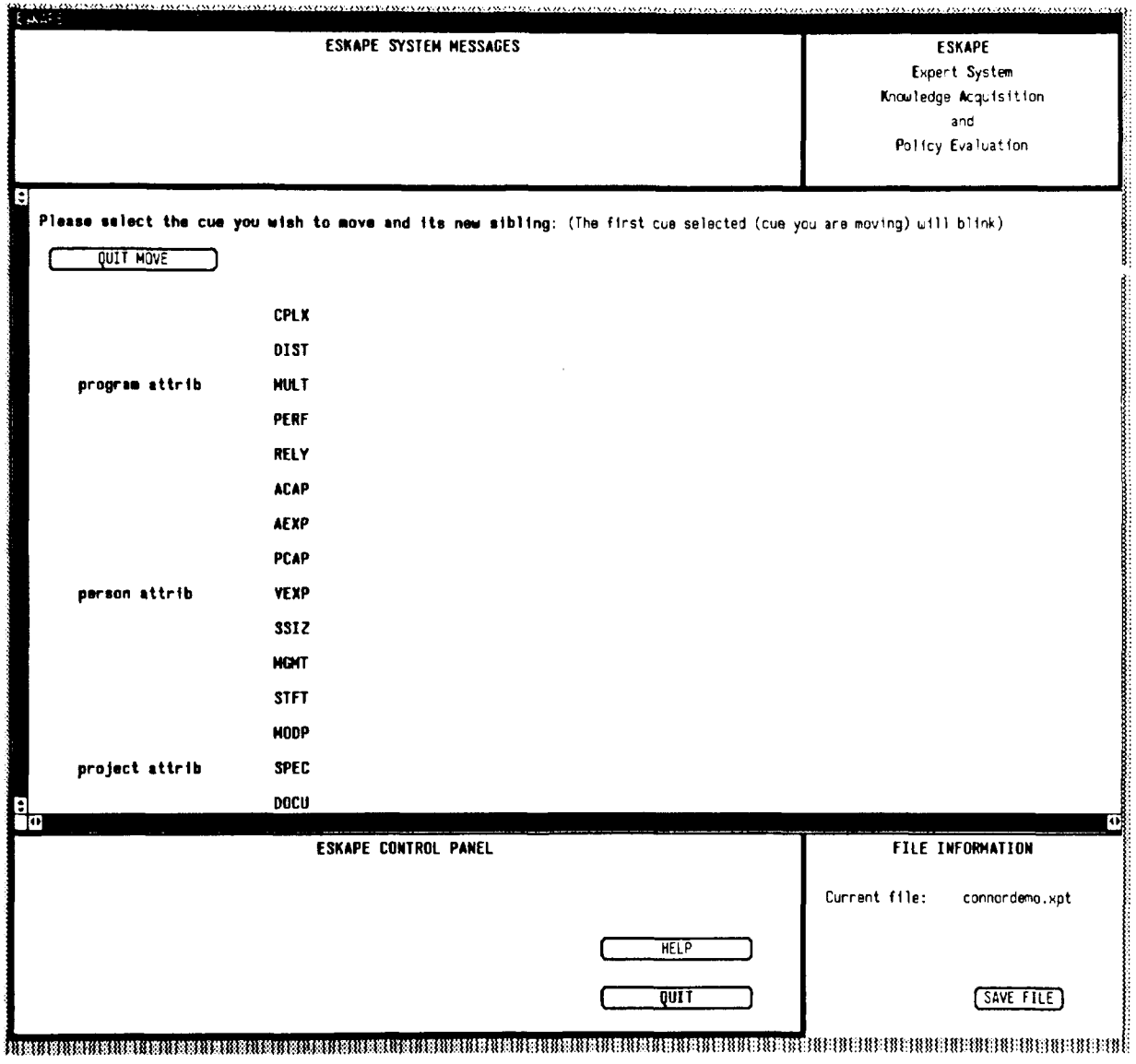

Figure 29. Move cue selection page.

presented. The user must first select the cue he or she desires to move. Once selected, that cue will flash. To deselect the first cue, simply click on the flashing cue when it is visible. As with all functions, context sensitive help is available as shown by Figure 30.

After the first cue has been selected, the user must select the location where the cue is to be moved. That location is specified by selecting a node which will be a sibling of the moved cue (similar to ADD CUE). Figure 31 illustrates the result when CPLX is moved from its location under *program attrib* to its new location as a sibling of STFT (under *project attrib*).

13. Saving the Knowledge Base

At periodic intervals, the expert should save the accumulated knowledge base. The user must exit all subfunctions and select the SAVE AS button the first time a knowledge base is saved. This allows the user to enter a meaningful 10 character file name as shown in Figure 32. Once the file is saved, the name is displayed in the File Information panel and the SAVE FILE button becomes visible as in Figure 33. Whenever the SAVE FILE button is visible, the current file may be saved under its current name at any time.

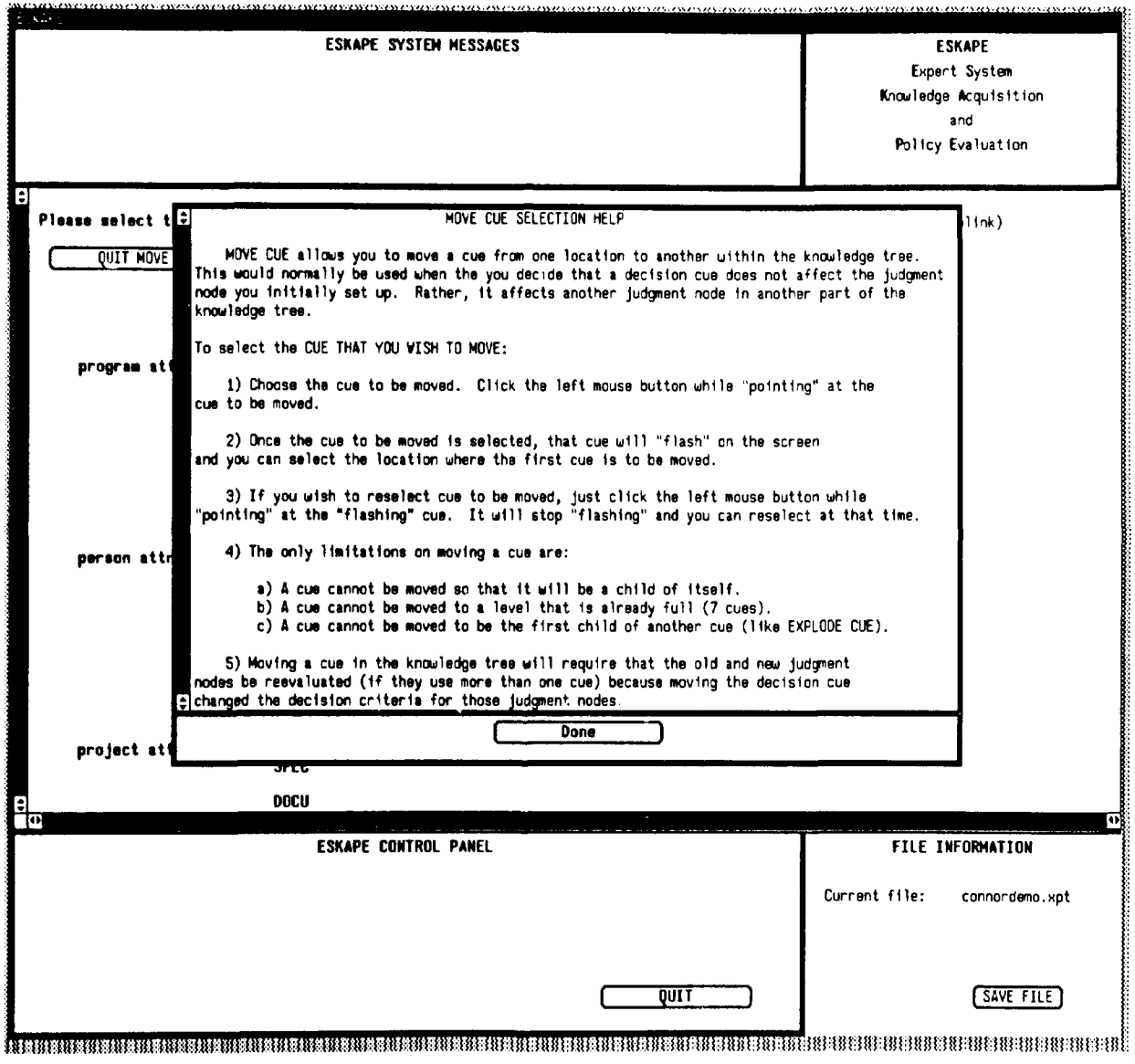

Figure 30. Move cue help screen.

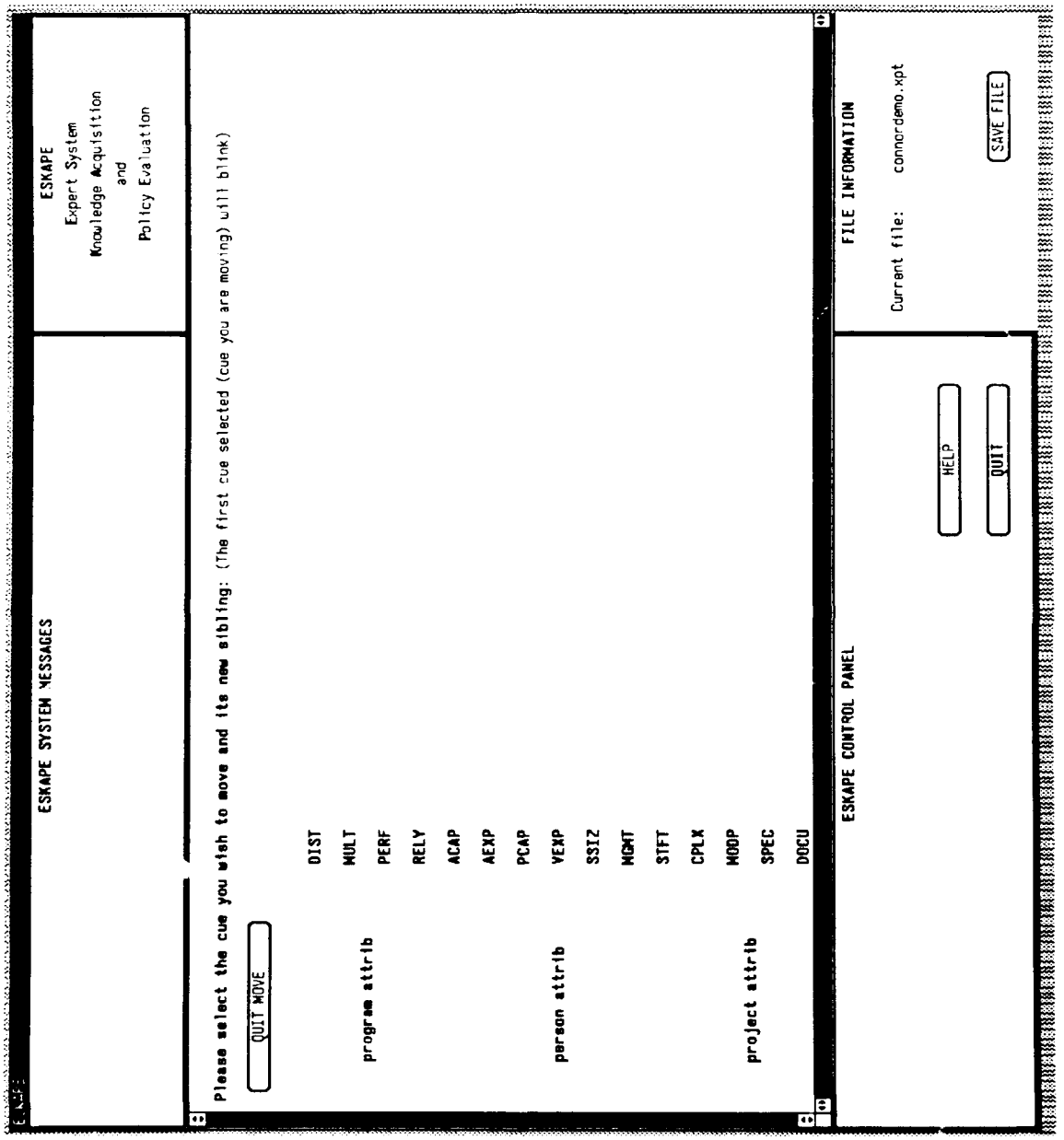

Figure 31. Move cue selection page after moving "CPLX."

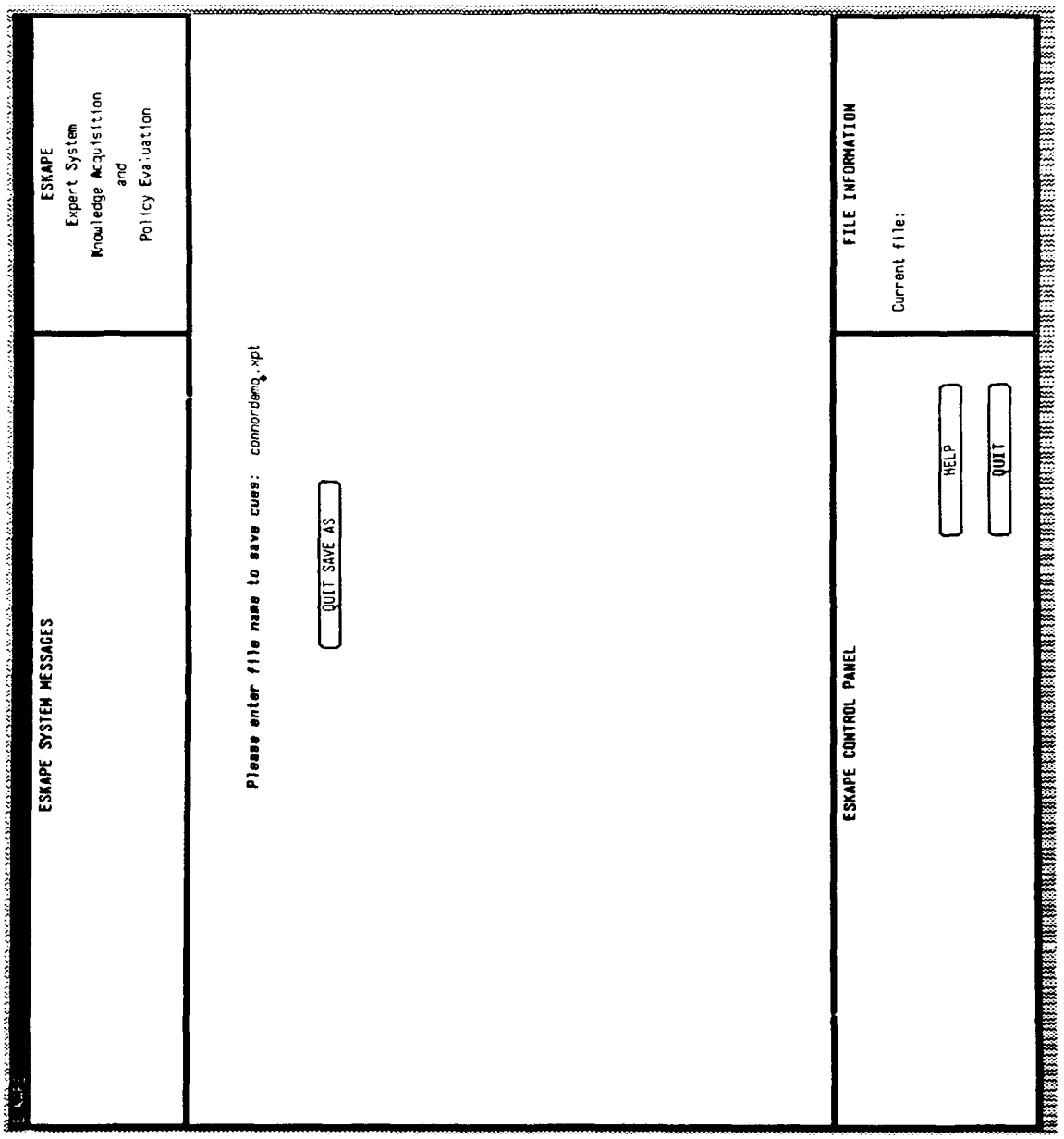

Figure 32. File "Save As" page.

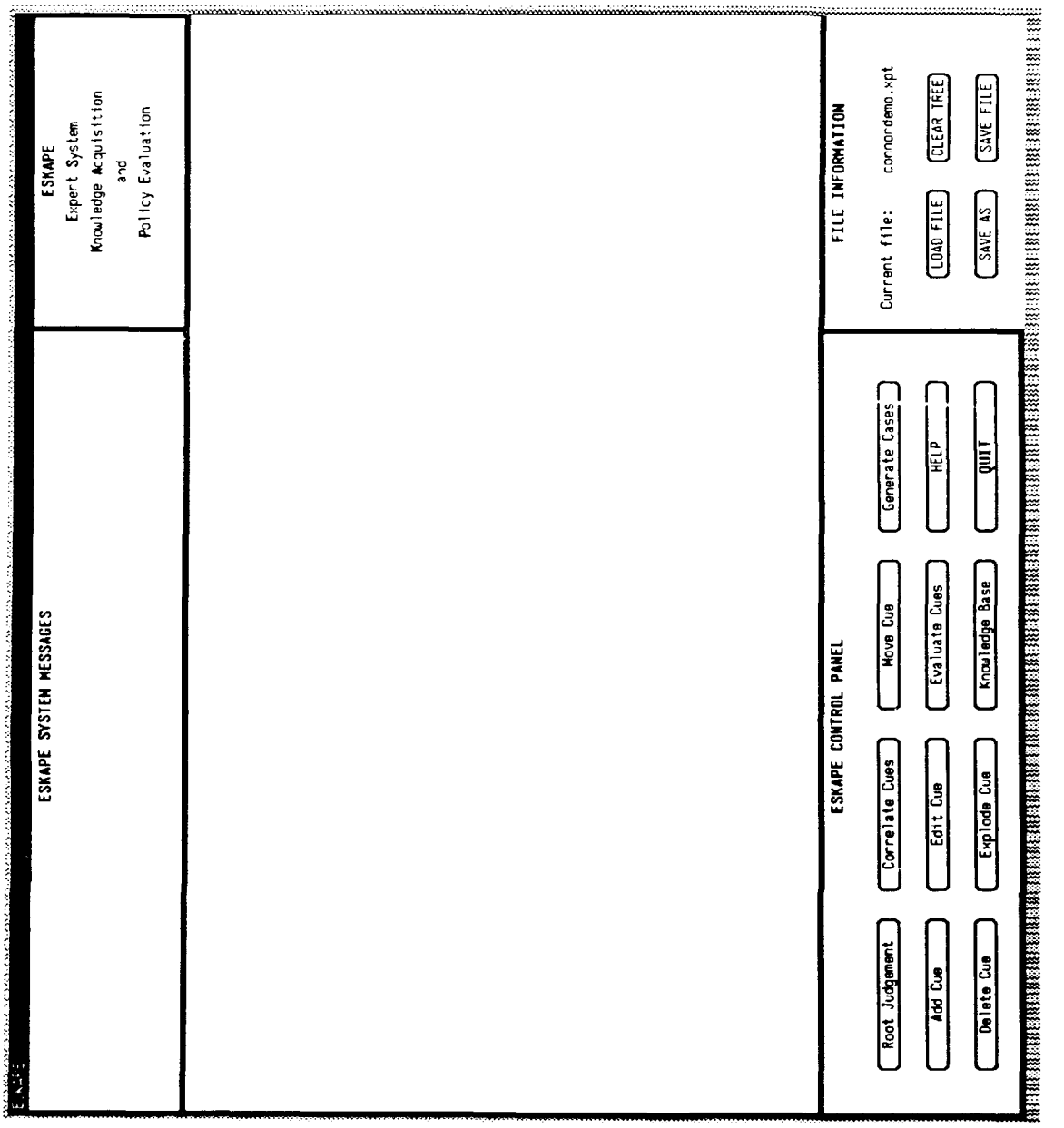

Figure 33. ESKAPE/CF screen showing "connordemo.xpt" loaded.

14. Loading Knowledge Bases

Once created and saved, a knowledge tree can be retrieved and loaded into memory for additional refinement. To retrieve an existing expert knowledge file, the user must select LOAD FILE in the File Information panel. After selection, the program searches for files with the ".xpt" extension and displays them for the user, as in figure 34. The user then selects the desired file to load into the ESKAPE program. If a knowledge tree already exists, the program will ask the user to confirm the file loading since it will overwrite the existing knowledge tree.

15. Clear Tree

The CLEAR TREE button on the File Information panel is used to erase the current knowledge tree. It is basically a mechanism by which an expert can clear a current session and start another without exiting the ESKAPE program.

C. SUMMARY

ESKAPE/CF is an interactive knowledge acquisition tool designed elicit relevant knowledge from an expert. The system provides experts with rapid feedback so that they can immediately evaluate their decision processes. This timely feedback overcomes the negative and counterproductive aspects of delayed feedback (Wickens 1984; Patterson 1990).

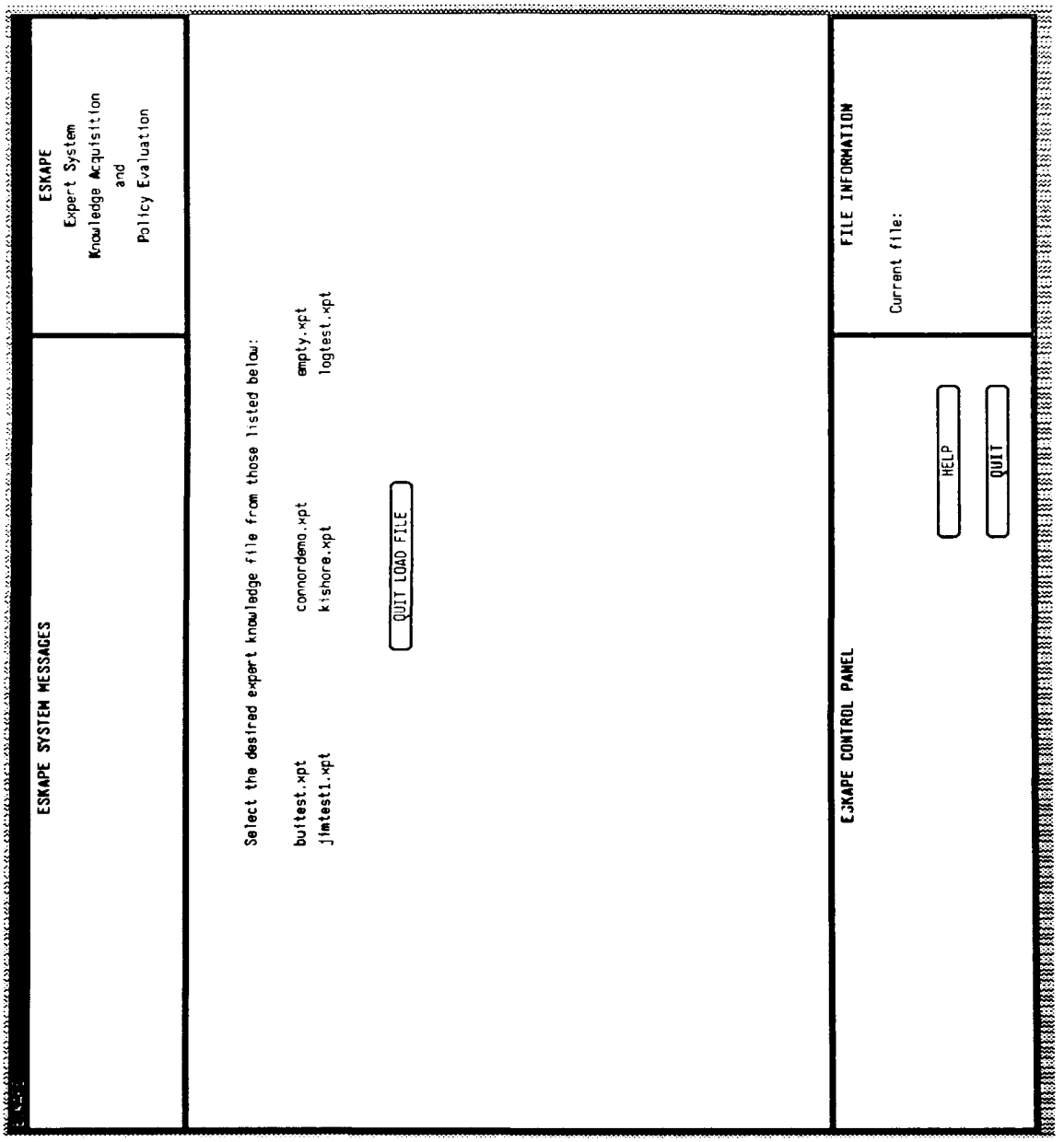

Figure 34. Load file selection page.

The expert can also expand and test an accumulated knowledge base without restrictions, thus minimizing any differences between the system and the user's cognitive style. Furthermore, by directly manipulating the system, the expert assumes many of the responsibilities currently performed by knowledge engineers. It is predicted that these two factors should result in a higher quality knowledge base that accurately represents the domain expert's decision processes. (Patterson 1990)

IV. CONCLUSIONS AND SUGGESTIONS FOR FUTURE RESEARCH

A. CONCLUSIONS

The demand for expert systems continues to grow at an increasing pace (Olson & Rueter 1987). Although the traditional method for creating a knowledge base has been knowledge engineering (Gruber 1988), the shortage of knowledge engineers coupled with the possibility of losing knowledge in the transfer process, has led to the use of interactive systems directly by the expert (Shaw & Gaines 1988). These systems have demonstrated the capability to shorten project development time (Boose 1985) and thus reduce the expert system construction "bottleneck."

Automated knowledge acquisition tools are supported by experts and expert system designers alike (Boose 1985). While usually interested and enthusiastic about a knowledge acquisition project (Boose 1985), an uncooperative expert will doom a project to failure (Olson & Rueter 1987). Gaining that expert's confidence and presenting a non-threatening computer interface are critical to project success. (Boose 1985)

Knowledge acquisition tools already in existence have had success at reducing the time required to create a knowledge base, but the psychological paradigms upon which they are

based may not offer the best method to capture "deep" policy knowledge (Boose 1985; Patterson 1990).

The ESKAPE/CF system is based on lens model techniques which have proven useful in eliciting "deep" policy knowledge from an expert (Balzer, Doherty, & O'Connor 1989). Coupling cognitive feedback techniques with an automated knowledge acquisition tool should not only reduce the time required for knowledge elicitation, but also capture knowledge that has been heretofore unattainable by existing systems and methods.

B. FUTURE RESEARCH

The development of the ESKAPE/CF system has revealed several areas where additional research is needed. Specifically,

1. ESKAPE/CF System Enhancements

a. Linear Logistic Analysis

The regression analysis module used to evaluate decision cues should be augmented by a linear logistic analysis module (Fienberg 1977). This will allow improved accuracy and reliability when evaluating categorical cues.

b. X-Windows Compatibility

The ESKAPE/CF system uses a proprietary interface (SunWindows) to generate system screens. If the system is modified to use an X-Windows interface, it would be more portable and possibly more widely used.

c. Applying ESKAPE/CF Knowledge to a Working Expert System

The knowledge captured by the ESKAPE/CF system must be ported to a working expert system if it is ever to be fully validated. Mechanisms and programs need to be developed to accomplish this task.

d. Expand the ESKAPE/CF Data Keeping Facilities

Currently, the only record-keeping functions are the knowledge base (tree) itself and the user trace file. Additional data that could be recorded are an expert's decision policies over time or comparisons of the decision policies of different experts for the same cognitive task.

2. Assessing ESKAPE/CF Validity

a. Applicable Task Areas

Empirical studies should be conducted to evaluate which knowledge acquisition tasks ESKAPE/CF is best suited.

b. System Usability

The system should be evaluated by "experts" to determine the usability of the ESKAPE/CF system. These evaluations would examine user interface issues along with the effectiveness of the model in capturing knowledge from the expert.

c. Comparison of Results using Different Experts

The results of the ESKAPE/CF sessions of different experts should be evaluated to determine if the system is

effective across several experts with different personalities and varying degrees of expertise.

d. Comparison of Results with Other Models

Once applicable task areas have been defined, the results of ESKAPE/CF knowledge acquisition sessions should be compared to those using other knowledge acquisition tools. The relative consistency and accuracy of the models would be of paramount concern. The system should be contrasted with existing automated tools such as Kitten and Aquinas (Boose & Bradshaw 1988; Shaw & Gaines 1987). Additionally, ESKAPE/CF cognitive feedback techniques could be evaluated against other learning paradigms such as Concept Learning System inductive techniques (Hunt, Marin, & Stone 1966; Quinlan 1983).

3. ESKAPE/CF Extensions

a. Learning Complex Tasks

Evaluate the ESKAPE/CF as a tool to teach complex decision making tasks. The underlying cognitive feedback mechanisms might be used to train individuals to better use the information available to them when making complex decisions.

b. Combining Expert Knowledge

Evaluate the system's effectiveness at combining the knowledge of several experts for a single decision task. (The prototype system will need to be enhanced to perform this analysis.)

c. Study the Decision Processes of Experts

Evaluate the utility of the ESKAPE/CF system in studying the decision processes of experts. Experts could be presented with varying sets of decision cues and their judgments analyzed to determine decision policies. This might also be used to provide experts with insight into previously unknown decision policies.

APPENDIX. SAMPLE USER TRACE

ESKAPE Version 1.0 User Action Trace File for Tue Feb 26 21:02:26 1991

```
02/26/91  21:02:26  GET  GET USER NAME   connor
02/26/91  21:02:40  LDF  LOAD FILE       select
02/26/91  21:02:53  LDF  LOAD FILE       connordemo.xpt
02/26/91  21:02:53  LDF  LOAD FILE       quit process
02/26/91  21:02:58  KBA  KNOWLEDGE BASE  select
02/26/91  21:03:02  KBA  KNOWLEDGE BASE  quit process
02/26/91  21:03:03  KBA  KNOWLEDGE BASE  select
02/26/91  21:03:34  KBA  KNOWLEDGE BASE  program attrib with Subtree
02/26/91  21:03:50  KBA  KNOWLEDGE BASE  program attrib with Subtree
02/26/91  21:04:06  KBA  KNOWLEDGE BASE  program attrib with Function
forms
02/26/91  21:04:25  KBA  KNOWLEDGE BASE  quit process
02/26/91  21:04:26  KBA  KNOWLEDGE BASE  select
02/26/91  21:04:28  KBA  KNOWLEDGE BASE  Knowledge tree
02/26/91  21:05:19  KBA  KNOWLEDGE BASE  quit process
02/26/91  21:05:21  EVA  EVALUATE CUE    select
02/26/91  21:05:24  EVA  EVALUATE CUE    program attrib
02/26/91  21:05:27  HLP  EVALUATE CUE    help screen 91
02/26/91  21:05:45  EVA  EVALUATE CUE    quit process
02/26/91  21:05:49  EVA  EVALUATE CUE    select
02/26/91  21:05:53  EVA  EVALUATE CUE    project attrib
02/26/91  21:06:21  EVA  EVALUATE CUE    project attrib with Subtree
02/26/91  21:06:28  EVA  EVALUATE CUE    project attrib with Stacked bars
02/26/91  21:06:38  EVA  EVALUATE CUE    project attrib with Function
forms
02/26/91  21:07:17  SAV  SAVE FILE       connordemo.xpt
02/26/91  21:07:28  EVA  EVALUATE CUE    select
02/26/91  21:07:32  EVA  EVALUATE CUE    size
02/26/91  21:07:38  EVA  EVALUATE CUE    quit process
02/26/91  21:07:46  SAV  SAVE FILE       connordemo.xpt
02/26/91  21:08:00  QUI  QUIT PROGRAM    quit process
```

LIST OF REFERENCES

Anderson, John R. 1982. Acquisition of Cognitive Skill. *Psychological Review* 89: 369-406.

Balzer, William K., Michael E. Doherty, and Raymond O'Connor, Jr. 1989. Effects of Cognitive Feedback on Performance. *Psychological Bulletin* 106: 410-433.

Boose, John H. 1985. A Knowledge Acquisition Program for Expert Systems based on Personal Construct Psychology. *International Journal of Man-Machine Studies* 23: 495-525.

Boose, John H. and Jeffrey M. Bradshaw. 1988. Expertise Transfer and Complex Problems: Using AQUINAS as a Knowledge-acquisition Workbench for Knowledge-based Systems. In *Knowledge-Based Systems*, vol. 2, *Knowledge Acquisition Tools for Expert Systems*, ed. J. H. Boose and B. R. Gaines, 39-64. San Diego: Academic Press.

Brehemer, B. 1979. Preliminaries to a Psychology of Inference. *Scandinavian Journal of Psychology* 20: 193-210.

Brunswick, E. 1955. The Conceptual Framework of Psychology. In *International Encyclopedia of Unified Science*, ed. O. Neurath, R. Carnap, and C Morris, 655-760. Chicago: University of Chicago Press.

Carroll, J. S. and J. W. Payne. 1976. The Psychology of Parole Decision Process: A Joint Application of Attribution Theory and Information-processing Psychology. In *Cognition and Social Behavior*, ed. J. S. Carroll and J. W. Payne. Hillsdale, NJ: Erlbaum.

Cooke, Nancy M. and James E. McDonald. 1988. The Application of Psychological Scaling Techniques to Knowledge Elicitation for Knowledge-based Systems. In *Knowledge-Based Systems*, vol. 2, *Knowledge Acquisition Tools for Expert Systems*, ed. J. H. Boose and B. R. Gaines, 65-82. San Diego: Academic Press.

Doherty, Michael E. and William K. Balzer. 1988. Cognitive Feedback. In *Human Judgement: The SJT View*, ed. Berndt Brehemer and C. R. B. Joyce, 163-188. Amsterdam: Elsevier Science Publishers.

Einhorn, Hillel J. 1974. Expert Judgment: Some Necessary Conditions and an Example. *Journal of Applied Psychology* 59: 562-571.

Einhorn, Hillel J., D. Kleinmuntz, and B. Kleinmuntz. 1979. Linear Regression and Process-Framing Models of Judgment. *Psychological Review* 86: 465-485.

Fienberg, Stephen E. 1977. *The Analysis of Cross-Classified Categorical Data*. Cambridge, MA: MIT Press.

Fitts, P. M. and M. I. Posner. 1967. *Human Performance*. Belmont, CA: Brooks Cole.

Gaines, Brian R. 1987. An overview of Knowledge-Acquisition and Transfer. *International Journal of Man-Machine Studies* 26: 453-472.

Garg-Janardan, Chaya and Gavriel Salvendy. 1988. A Conceptual Framework for Knowledge Elicitation. In *Knowledge-Based Systems*, vol. 2, *Knowledge Acquisition Tools for Expert Systems*, ed. J. H. Boose and B. R. Gaines, 119-129. San Diego: Academic Press.

Gruber, Thomas R. 1988. Acquiring Strategic Knowledge from Experts. *International Journal of Man-Machine Studies* 29: 579-597.

Hammond, K. R., G. H. McCleland, and J. Mumpower. 1980. *Human Judgment and Decision Making: Theory, Methods and Procedures*. New York: Praeger.

Hoffman, Robert R. 1989. A Survey of Methods for Eliciting the Knowledge of Experts. *SIGART Newsletter* 108 (April).

Hunt, E. B., J. Marin, and P. T. Stone. 1966. *Experiments in Induction*. New York: Academic Press.

Kelly, G. A. 1955. *The Psychology of Personal Constructs*. New York: Norton.

Kitto, Catherine M. and John H. Boose. 1989. Selecting Knowledge Acquisition Tools and Strategies Based on Application Characteristics. *International Journal of Man-Machine Studies* 31: 149-160.

Larkin, J. H., J. McDermott, D. P. Simon and H. A. Simon. 1980. Expert and Novice Performance in Solving Physics Problems. *Science* 208: 1335-1342.

Libby, R. 1981. *Accounting and Human Information Processing: Theory and Applications.* Englewood Cliffs, NJ: Prentice Hall.

McGraw, K. L. and A. Riner. 1987. Task Analysis: Structuring the Knowledge Acquisition Process. *TI Technical Journal* November-December: 16-21.

McNurlin, Barbara C. and Ralph H Sprague. 1989. *Information Systems Management in Practice.* Englewood Cliffs, NJ: Prentice Hall.

Meehl, P. 1954. *Clinical Versus Statistical Prediction: Theoretical Analysis and a Review of the Evidence.* Minneapolis: University of Minnesota Press.

Miller, G. A. 1956. The Magical Number Seven Plus or Minus Two: Some limits on our Capacity for Processing Information. *Psychological Review* 63: 81-97.

Olson, Judith Reitman and Henry H. Rueter. 1987. Extracting Expertise from Experts: Methods for Knowledge Acquisition. *Expert Systems* 4: 152-168.

Patterson, Charles. 1990. *Cognitive Feedback as a Tool for Knowledge Acquisition.* Master's Thesis, Naval Postgraduate School, Monterey, CA.

Phelps, Ruth H. and James Shanteau. 1978. Livestock Judges: How Much Information Can an Expert Use? *Organizational Behavior and Human Performance* 21: 209-217.

Press, W., B. Flannery, S. Teukolsky, and W. Vetterling. 1990. *Numerical Recipes in C.* Cambridge, MA: Cambridge University Press.

Quinlan, J. R. 1983. Learning Efficient Classification Procedures and Their Application to Chess End Games. In *Machine Learning: An Artificial Intelligence Approach*, ed. R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. Palo Alto, CA: Tioga Publishing.

Shanteau, James. 1990. The Psychology of Experts: An Alternative View. *Organizational Behavior and Human Performance*, forthcoming.

Shanteau, James. 1988. Psychological Characteristics and Strategies of Expert Decision Makers. *Acta Psycologica* 68: 203-215.

Shanteau, James and G. Gaeth. 1981. *Evaluation of the Field Method of Soil Texture Classification: A Psychological Analysis of Accuracy and Consistency.* Tech. Rep. 79-1. Kansas State University, Department of Psychology.

Shaw, Mildred L. G. and Brian R. Gaines. 1988. KITTEN: Knowledge Initiation and Transfer Tools for Experts and Novices. *International Journal of Man-Machine Studies* 27: 251-280.

Stewart, Thomas R. 1988. Judgment Analysis: Procedures. In *Human Judgement: The SJT View*, ed. Berndt Brehemer and C. R. B. Joyce, Amsterdam: Elsevier Science Publishers.

Vincinanza, Stephen S. 1990. Case-Based Software Cost Estimation. Ph.D. diss., Carnegie Mellon University.

von Winterfeldt, Detlof. 1988. Expert Systems and Behavioral Decision Research. *Decision Support Systems* 4: 461-471.

Wickens, C. D. 1984. *Engineering Psychology and Human Performance.* Columbus, OH: Charles E. Merrill.